

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня «бакалавр»  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка веб-сайту для пошуку роботи та  
працівників»**

Виконав студент групи ІПЗ-20-3  
БАРАНОВСЬКИЙ Даніїл Сергійович

Керівник роботи:  
КУБРАК Юрій Олександрович

Консультант:  
МОРОЗОВ Андрій Васильович

Рецензент:  
ЄФРЕМОВ Юрій Миколайович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**«ЗАТВЕРДЖУЮ»**

Зав. кафедри інженерії  
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2024 р.

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**

Здобувач вищої освіти: **БАРАНОВСЬКИЙ Даніїл Сергійович**

Керівник роботи: **КУБРАК Юрій Олександрович**

Тема роботи: **«Розробка веб-сайту для пошуку роботи та працівників»**,  
затверджена Наказом закладу вищої освіти від «23» лютого 2024 р., №74с

Вихідні дані для роботи: **Веб-сайт для пошуку роботи та працівників як  
інструмент взаємодії між роботодавцями та працівниками на ринку праці.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що  
їх стосуються:

| Розділ | Консультант  | Завдання видав | Завдання прийняв |
|--------|--------------|----------------|------------------|
| 1      | Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |
| 2      | Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |
| 3      | Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |

## РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаної літератури та додатків. Текстова частина викладена на 70 сторінках друкованого тексту. Джерела інформації містять 20 найменувань і займають 2 сторінки. Пояснювальна записка має 7 сторінок додатків. У роботі наведено 56 рисунків. Загальний обсяг роботи – 79 сторінки.

У першому розділі було виконано аналіз напрямків використання інформаційних технологій для реалізації веб-сайту для пошуку роботи та працівників. Обґрунтовано постановку задачі, проаналізовано аналоги, вибрано архітектуру та засоби реалізації, а також визначено вимоги до апаратного забезпечення

У другому розділі визначено варіанти використання системи та створено об'єктно-орієнтовану структуру, що включає ключові компоненти та їх взаємодії. Також розроблено структуру бази даних, враховуючи потреби зберігання та обробки інформації про роботу та працівників. На додаток, виконано проектування та реалізацію алгоритмів, необхідних для ефективної роботи системи.

У третьому розділі описано процес встановлення та налаштування системи, включаючи детальну конфігурацію всіх компонентів платформи. Розглянуто структуру інтерфейсу веб-сайту, що забезпечує зручну та інтуїтивно зрозумілу взаємодію користувачів із системою. Також детально описано мануальне тестування веб-сайту, яке дозволило виявити та усунути помилки.

Висновки містять узагальнені результати проведених досліджень та розробки. У них висловлено основні висновки з кожного розділу та підсумки роботи в цілому.

Ключові слова: PHP, Symfony, Vue.js, MySQL, MVC.

|           |      |                  |        |      |                                                                       |                                           |      |        |
|-----------|------|------------------|--------|------|-----------------------------------------------------------------------|-------------------------------------------|------|--------|
|           |      |                  |        |      | ІПЗ.КР.Б-121-24-ПЗ                                                    |                                           |      |        |
| Змн.      | Арк. | № докум.         | Підпис | Дата |                                                                       |                                           |      |        |
| Розроб.   |      | Барановський Д.С |        |      | Пояснювальна записка<br><br>Веб-сайт для пошуку роботи та працівників | Літ.                                      | Арк. | Аркуші |
| Керівник  |      | Кубрак Ю.О.      |        |      |                                                                       |                                           | 3    | 79     |
|           |      |                  |        |      |                                                                       |                                           |      |        |
| Рецензент |      | Єфремов Ю.М      |        |      |                                                                       |                                           |      |        |
| Затвердив |      | Вакалюк Т.А      |        |      |                                                                       | Житомирська політехніка<br>група ІПЗ-20-3 |      |        |

## ABSTRACT

The bachelor's thesis consists of an introduction, three chapters, conclusions, a list of references and appendices. The text part is laid out on 70 pages of printed text. Information sources contain 20 items and occupy 2 pages. The explanatory note has 7 pages of annexes. The work contains 56 drawings. The total volume of work is 79 pages.

In the first section, an analysis of the directions of using information technologies for the implementation of a website for job and employee search was carried out. The statement of the problem is substantiated, analogs are analyzed, the architecture and means of implementation are selected, and the requirements for the hardware are determined

In the second section, the use cases of the system are defined and an object-oriented framework is created, which includes key components and their interactions. The structure of the database was also developed, taking into account the needs of storing and processing information about work and employees. In addition, the design and implementation of the algorithms necessary for the efficient operation of the system was performed.

The third section describes the system installation and configuration process, including detailed configuration of all platform components. The structure of the website interface, which ensures convenient and intuitive user interaction with the system, is considered. Also detailed is the manual testing of the website, which allowed us to identify and eliminate errors.

The conclusions contain the generalized results of the conducted research and developments. They express the main conclusions from each section and summaries work in general.

Keywords: PHP, Symfony, Vue.js, MySQL, MVC.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 4    |

## ЗМІСТ

|                                                                                                                                        |    |
|----------------------------------------------------------------------------------------------------------------------------------------|----|
| РЕФЕРАТ .....                                                                                                                          | 3  |
| ABSTRACT .....                                                                                                                         | 4  |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....                                                                                                        | 6  |
| ВСТУП.....                                                                                                                             | 7  |
| РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ<br>ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-САЙТУ ДЛЯ ПОШУКУ РОБОТИ ТА<br>ПРАЦІВНИКІВ ..... | 9  |
| 1.1 Опис проекту та постановка задачі.....                                                                                             | 9  |
| 1.2 Аналіз аналогів програмного продукту.....                                                                                          | 11 |
| 1.3 Вибір архітектури веб-сайту для пошуку роботи та працівників.....                                                                  | 19 |
| 1.4 Обґрунтування вибору засобів реалізації та вимоги до апаратного забезпечення .                                                     | 22 |
| Висновки до першого розділу .....                                                                                                      | 25 |
| РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ ДЛЯ ПОШУКУ РОБОТИ ТА<br>ПРАЦІВНИКІВ .....                                                             | 27 |
| 2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи .                                                     | 27 |
| 2.2 Розробка бази даних системи .....                                                                                                  | 35 |
| 2.3 Проектування та реалізація алгоритмів роботи системи.....                                                                          | 41 |
| 2.4 Реалізація функціоналу веб-сайту для пошуку роботи.....                                                                            | 46 |
| Висновки до другого розділу.....                                                                                                       | 50 |
| РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-САЙТОМ ДЛЯ ПОШУКУ<br>РОБОТИ ТА ПРАЦІВНИКІВ .....                                           | 52 |
| 3.1 Порядок встановлення та налаштування параметрів системи .....                                                                      | 52 |
| 3.2 Структура інтерфейсу веб-сайту для пошуку роботи та працівників .....                                                              | 54 |
| 3.3 Тестування роботи веб-сайту для пошуку роботи та працівників.....                                                                  | 66 |
| Висновки до третього розділу .....                                                                                                     | 69 |
| ВИСНОВКИ .....                                                                                                                         | 70 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                                       | 71 |
| ДОДАТКИ .....                                                                                                                          | 73 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface (Інтерфейс програмування додатків), набір засобів і протоколів для створення програмного забезпечення

CV – Curriculum Vitae (Резюме), документ, що містить інформацію про досвід роботи, освіти та навички працівника

MVC – Model-View-Controller (Модель-Вид-Контролер), архітектурний шаблон для побудови програмного забезпечення

SQL – Structured Query Language (Структурована мова запитів), мова для управління даними в реляційних базах даних

UI – User Interface (Інтерфейс користувача), частина системи, з якою взаємодіє користувач

UX – User Experience (Користувацький досвід), загальний досвід користувача під час взаємодії з системою

БД – База даних

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 6    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## ВСТУП

**Актуальність теми:** у наш час актуальність теми "Розробка веб-сайту для пошуку роботи та працівників" в контексті випускної кваліфікаційної роботи визначається кількома ключовими факторами. Зростання ринкової конкуренції, стрімкий розвиток інформаційних технологій та необхідність ефективного використання людських ресурсів створюють особливо важливий контекст для подальшого дослідження та розробки.

В сучасному суспільстві, де пошук роботи та вакансії є не лише актуальними, але й визначають кар'єрний шлях багатьох людей, розробка веб-сайту для покриття цих потреб стає надзвичайно важливою задачею.

Зростання кількості користувачів Інтернету та їхній високий рівень взаємодії з онлайн-ресурсами перетворюють сучасні веб-сайти не лише на інформаційні, але і на активні інструменти для взаємодії між роботодавцями та працівниками.

Розробка веб-сайту для пошуку роботи та працівників не лише спрощує та прискорює процес підбору кандидатів та знаходження відповідних вакансій, але й створює комфортне середовище для взаємодії між учасниками ринку праці.

Враховуючи тенденції ринку праці, де важливо не лише швидко, але і якісно вибрати підходящого працівника або роботу, застосування сучасних методів розробки веб-сайтів стає необхідною частиною проекту. Використання відповідних технологій та аналітичних інструментів може покращити ефективність вибору, сприяти оптимізації процесів та забезпечити максимальну взаємодію між обома сторонами – роботодавцями та працівниками.

Отже, ця тема кваліфікаційної роботи є актуальною, оскільки її дослідження спрямоване на розвиток ефективних та сучасних підходів до розробки веб-сайту для пошуку роботи та працівників. Враховуючи велике значення онлайн-платформ у сфері працевлаштування, цей проект може покращити якість та ефективність взаємодії на ринку праці.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 7    |

**Мета роботи:** Розробка ефективного та сучасного веб-сайту для пошуку роботи та працівників, який відповідає сучасним тенденціям ринку праці та вимогам користувачів.

**Завдання роботи:**

- Вивчення актуальних тенденцій на ринку праці та аналіз вимог роботодавців та працівників до платформ для пошуку роботи.
- Розробка інноваційних функцій та інтерфейсу веб-сайту для забезпечення зручного та ефективного взаємодії між учасниками ринку праці.
- Використання сучасних технологій розробки веб-сайтів та аналітичних інструментів для оптимізації процесів вибору кандидатів та знаходження вакансій.

**Об'єкт дослідження:** Веб-сайт для пошуку роботи та працівників як інструмент взаємодії між роботодавцями та працівниками на ринку праці.

**Методи дослідження:** Аналіз сучасних підходів до розробки веб-сайтів та їхніх застосувань у сфері працевлаштування.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 8    |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

# РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-САЙТУ ДЛЯ ПОШУКУ РОБОТИ ТА ПРАЦІВНИКІВ

## 1.1 Опис проекту та постановка задачі

У сучасному світі, коли ринок праці швидко змінюється та розвивається, існує велика потреба в ефективних засобах для пошуку роботи та працівників. З урахуванням цієї потреби, мій проект спрямований на розробку та реалізацію веб-сайту, призначеного для полегшення процесу знаходження роботи та працівників.

Головною задачею цього проекту є створення інноваційного веб-сайту, який надасть зручний і ефективний інструмент для спілкування між роботодавцями та працівниками. Проект розробляється з урахуванням сучасних тенденцій у сфері технологій та дозволить користувачам максимально ефективно взаємодіяти на ринку праці.

Для успішної реалізації проекту необхідно вирішити ряд ключових завдань, які визначаються як основні виклики та пріоритети в рамках цього проекту. Завдання розподілені за функціональними блоками та об'єднані спільним завданням або метою.

### **Розробка інтерфейсу користувача (UI/UX):**

Створення зручного інтерфейсу для користувачів:

- Проаналізувати потреби цільової аудиторії.
- Розробити інтуїтивно зрозумілу навігацію.
- Забезпечити адаптивність до різних пристроїв та розмірів екранів.

Дизайн та оформлення:

- Розробити естетичний та привабливий дизайн.
- Визначити кольорову палітру та шрифти.
- Забезпечити легкість сприйняття інформації.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 9    |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

## **Функціональність веб-сайту:**

### **Реєстрація та авторизація:**

- Розробити систему реєстрації для працівників та роботодавців.
- Забезпечити безпеку та конфіденційність даних.

### **Пошук та фільтрація вакансій і профілів працівників:**

- Реалізувати ефективний механізм пошуку вакансій та працівників.
- Дозволити користувачам встановлювати фільтри за різними критеріями.

### **Створення та редагування профілю:**

- Дати можливість працівникам та роботодавцям змінювати інформацію про свій профіль, контакти та компанію
- Забезпечити можливість завантаження документів (CV).

## **База даних та зв'язок між користувачами:**

### **Створення бази даних:**

- Розробити структуру бази даних для зберігання інформації про користувачів, вакансії, відгуки та пропозиції.

### **Модуль сповіщень:**

- Забезпечити сповіщення про дії працівників, або роботодавців з відгуками, або пропозиціями.

## **Тестування та вдосконалення:**

### **Тестування веб-сайту:**

- Провести тестування на різних етапах розробки (юзабіліті, функціональне, відмовостійкість).
- Виправити виявлені помилки та недоліки.

### **Оптимізація продуктивності:**

- Вдосконалити швидкодію та продуктивність веб-сайту.
- Забезпечити оптимальне використання ресурсів сервера.

Цей перелік задач є фундаментом для успішної реалізації випускної роботи та створення веб-сайту, який відповідає вимогам та потребам користувачів.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 10   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## 1.2 Аналіз аналогів програмного продукту

Так як тема моєї випускної кваліфікаційної роботи - це розробка веб-сайту для пошуку роботи та працівників. Важливою частиною розробки є аналіз аналогів різноманітних веб-сайтів які по функціоналу відповідають моїй темі, оскільки їх кількість на українському ринку доволі велика, більшість існуючих продуктів вже зарекомендували себе як хороші та ефективні рішення, і мало хто ризикує конкурувати з ними.

Мій підхід полягатиме в аналізі та дослідженні найкращих реалізацій веб-сайтів у цьому сегменті. І я вирішив одразу встановити високу планку для свого продукту, оцінюючи якісні аналоги та різноманітні підходи до розробки. Основний фокус буде зосереджений на популярних продуктах українського ринку з метою виявлення їхніх переваг і можливостей для подальшого використання у власному проекті.

Це дозволить мені заздалегідь визначити стандарти та вимоги, що є ключовими для розробки успішного веб-сайту в області пошуку роботи та працівників, а також допоможе визначити ніші та можливості для вдосконалення та конкуренції на ринку.

Почну я з найбільш якісного продукту, на мою думку: це Work.ua. Цей веб-сайт вже довгий час займає лідерські позиції на українському ринку та зарекомендував себе як надійний і ефективний інструмент для пошуку роботи та найму працівників.

Що стосується його інтерфейсу та користувацької дружельності, Work.ua вражає простотою та лаконічністю. Чітка структура сайту та інтуїтивно зрозумілі розділи дозволяють користувачам легко орієнтуватися та швидко знаходити потрібну інформацію. Крім того, розробники Work.ua продумано впровадили систему фільтрів та пошукових опцій, що дозволяє точно визначити вакансії чи кандидатів, відповідно до потреб користувача.

Однією з ключових переваг Work.ua є активна спільнота користувачів та регулярне оновлення вакансій. Система рекомендацій, базована на

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 11   |

індивідуальних профілях, робить процес пошуку ще ефективнішим та сприяє швидкому знаходженню відповідних пропозицій.

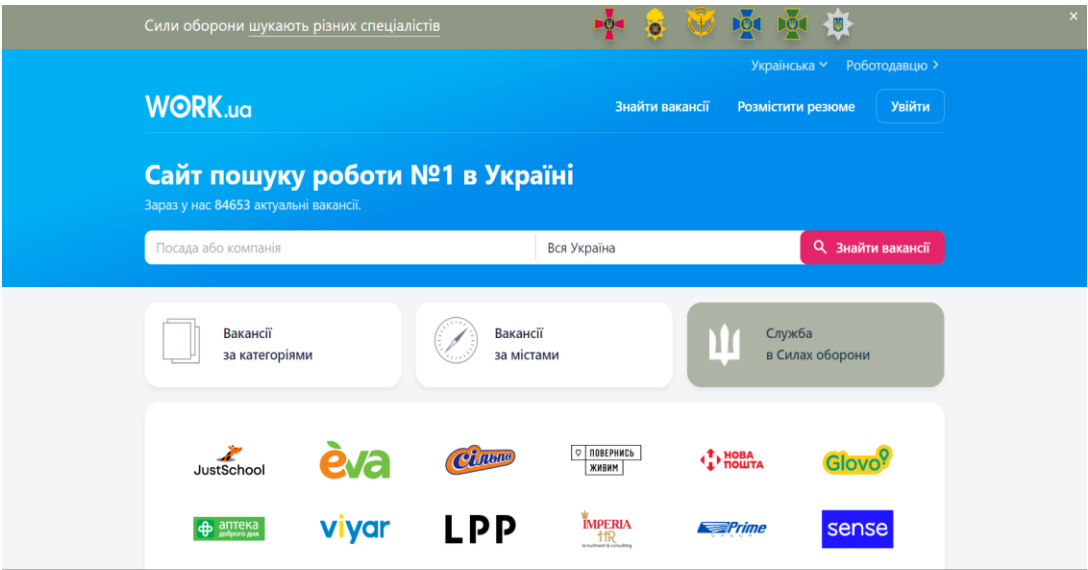


Рис. 1.1 - Головна сторінка сайту Work.ua

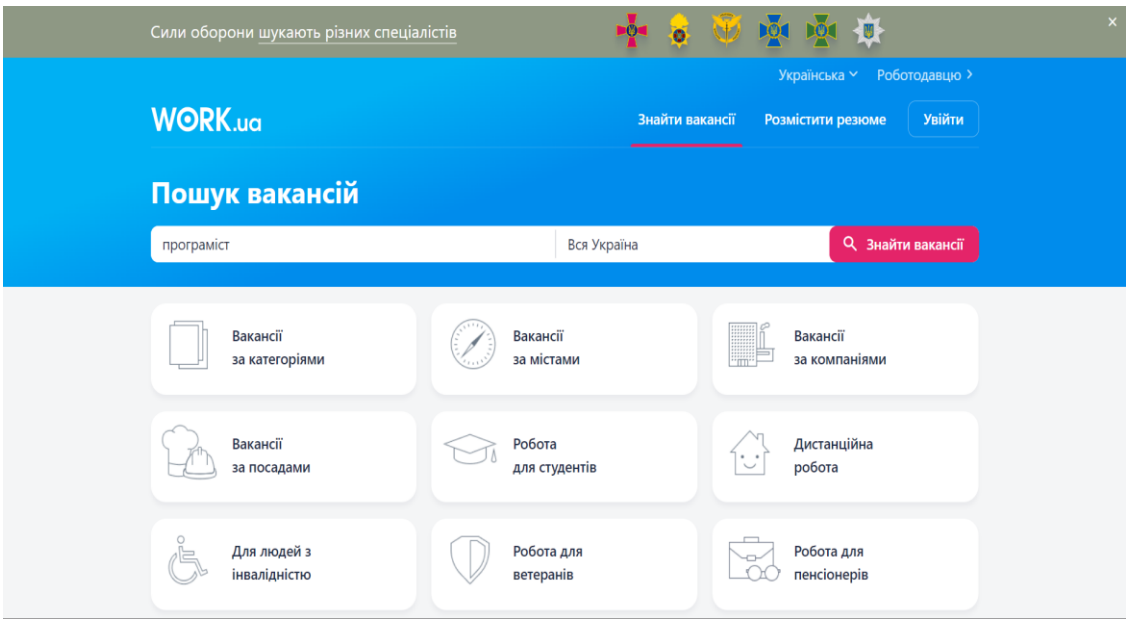


Рис. 1.2 - Сторінка пошуку вакансій сайту Work.ua

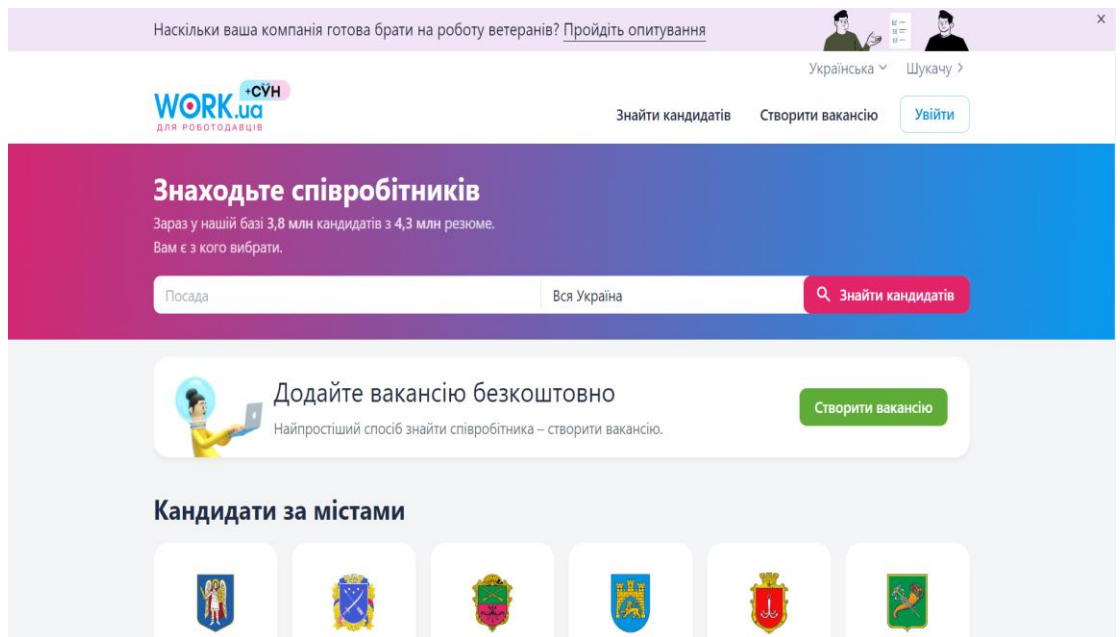


Рис. 1.3 - Головна сторінка сайту Work.ua для роботодавців

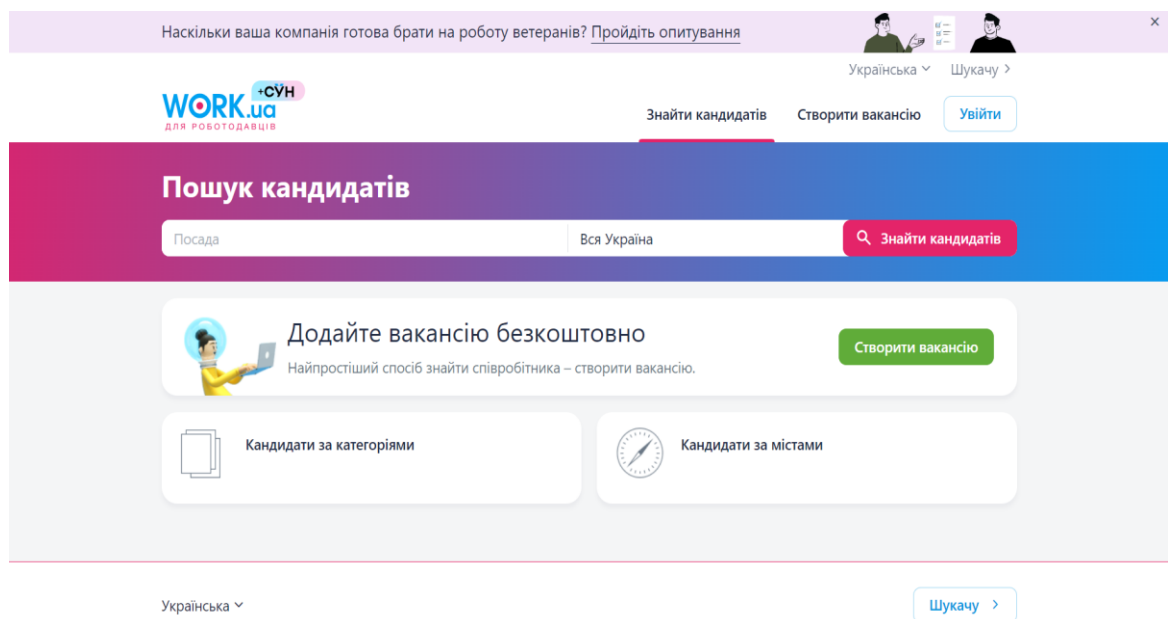


Рис. 1.4 - Сторінка пошуку працівників сайту Work.ua для роботодавців

Наступним розглянемо сайт Rabota.ua, що є ще однією визнаною та популярною платформою для пошуку роботи та працівників на українському ринку праці.

Rabota.ua також володіє рядом переваг, які варто врахувати під час аналізу конкурентів. Перш за все, варто відзначити деталізацію та обширність інформації на сайті. Кожна вакансія чи резюме мають докладний опис, а також

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 13   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

можливість прикріплення файлів, що сприяє більш повному представленню інформації сторонами.

Ще однією ключовою особливістю є акцент на вивченні ринку праці. Rabota.ua надає користувачам статистику щодо попиту та пропозиції на ринку праці, що може бути корисним при прийнятті рішень щодо вибору кар'єрного шляху або найму персоналу.

Також слід відзначити технологічні нововведення, що використовуються на сайті. Наприклад, система сповіщень та оновлень, яка дозволяє користувачам отримувати актуальну інформацію безпосередньо на електронну пошту чи мобільний додаток.

Важливо враховувати і систему рейтингів та відгуків, яка є ефективним інструментом для оцінки якості вакансій та працедавців. Це створює додатковий рівень довіри серед користувачів.

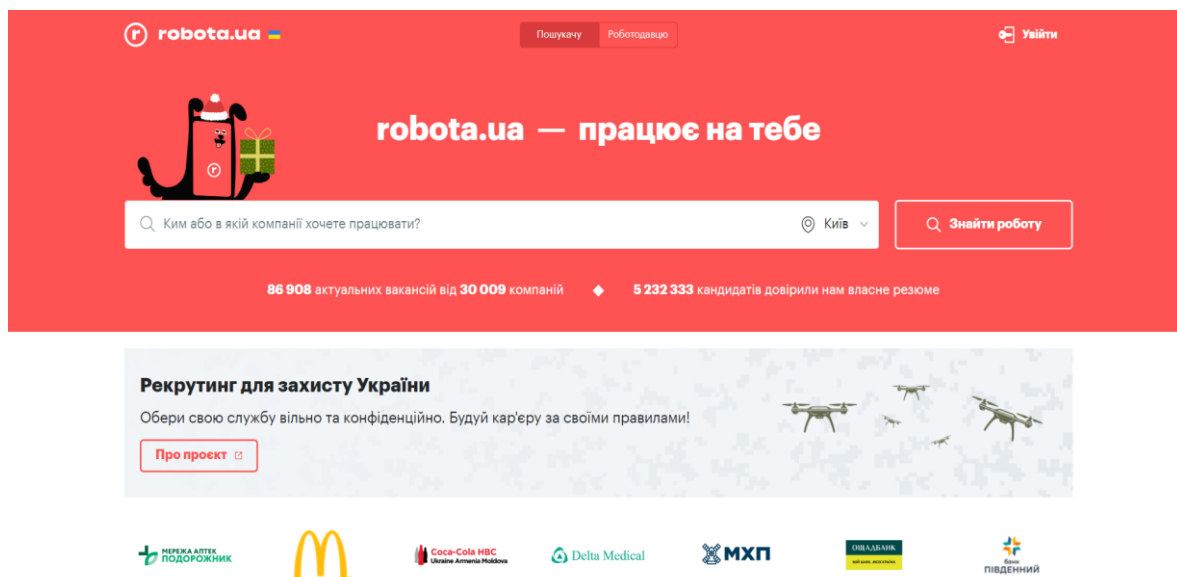


Рис. 1.5 - Головна сторінка сайту Rabota.ua

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 14   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

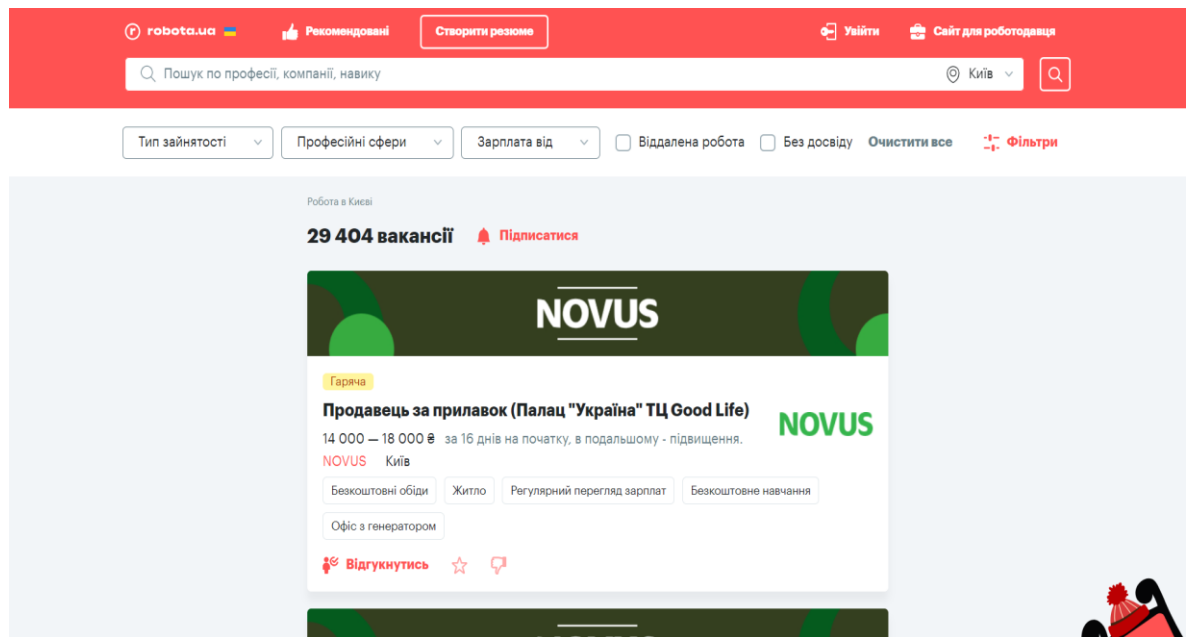
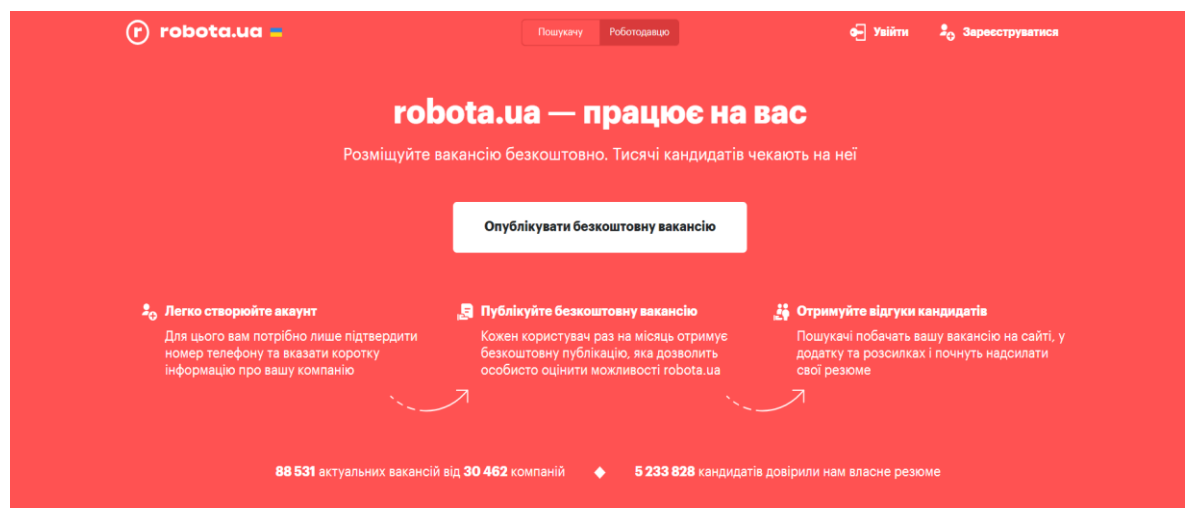


Рис. 1.6 - Сторінка пошуку вакансій сайту Rabota.ua



**5 мільйонів кандидатів чекають на вас в базі резюме від robota.ua**

Рис. 1.7 - Головна сторінка сайту Rabota.ua для роботодавців

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 15   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

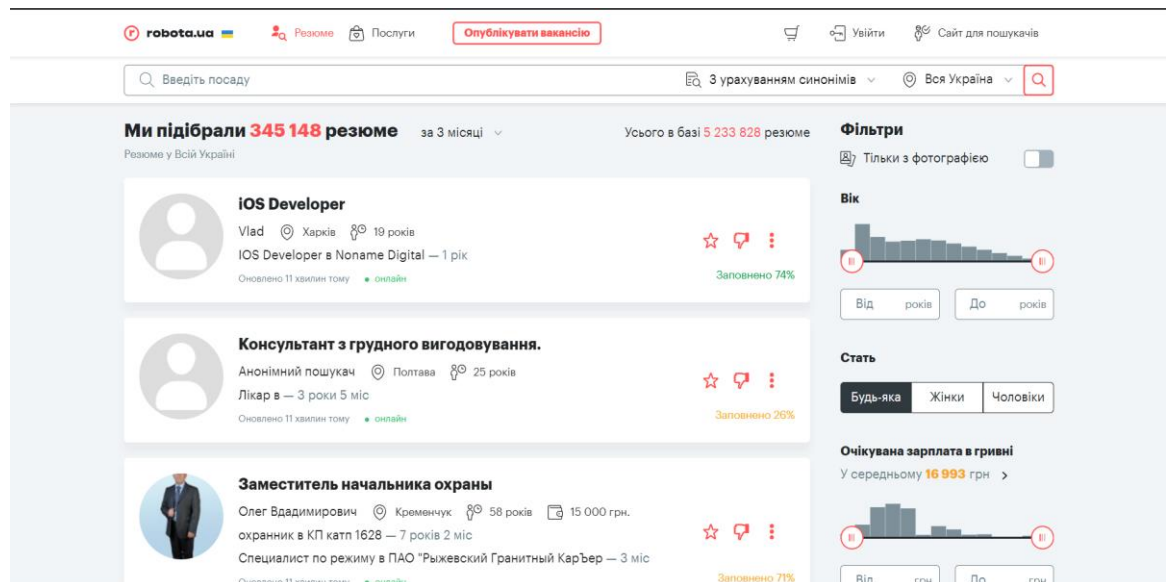


Рис. 1.8 - Сторінка пошуку працівників сайту Rabota.ua для роботодавців

Третім сайтом для аналізу буде Jooble, який також входить в число важливих учасників українського ринку праці та пропонує послуги з пошуку роботи та працівників.

Характерною особливістю Jooble є його концепція агрегатора вакансій, яка об'єднує в собі пропозиції різних роботодавців з різних джерел. Такий підхід дозволяє користувачам отримати ширший огляд робочих місць на ринку праці та економити час при пошуку.

Один з сильних боків Jooble - це масштабність та географічна доступність. Сервіс працює не лише на українському, а й на міжнародному рівні, що дає користувачам можливість знаходження роботи за кордоном або найму міжнародних працівників.

Інтерфейс сайту Jooble є достатньо простим та зрозумілим. Сучасний дизайн та зручна навігація дозволяють швидко зорієнтуватися на сайті. Крім того, наявність різноманітних фільтрів та опцій пошуку дозволяє користувачам точно налаштувати свої критерії та отримати релевантні результати.

Важливо враховувати і функціонал сторінок компаній, де роботодавці можуть детально представити інформацію про себе, свою діяльність та вакансії, що дозволяє створювати повніші та привабливі об'яви.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 16   |

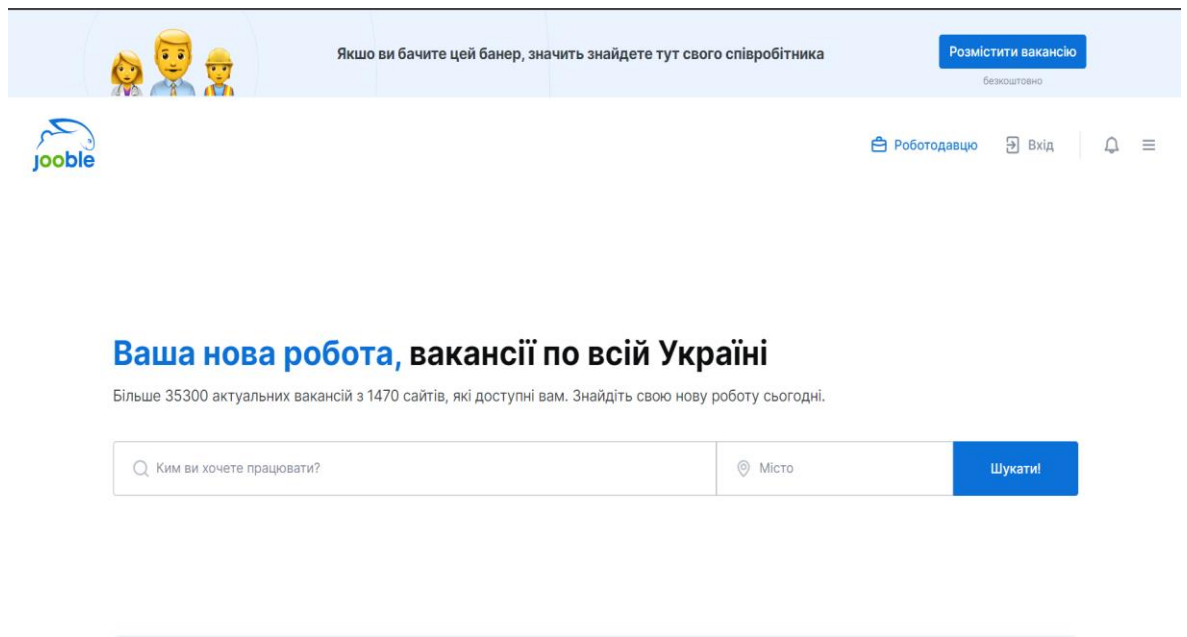


Рис. 1.9 - Головна сторінка сайту Jooble

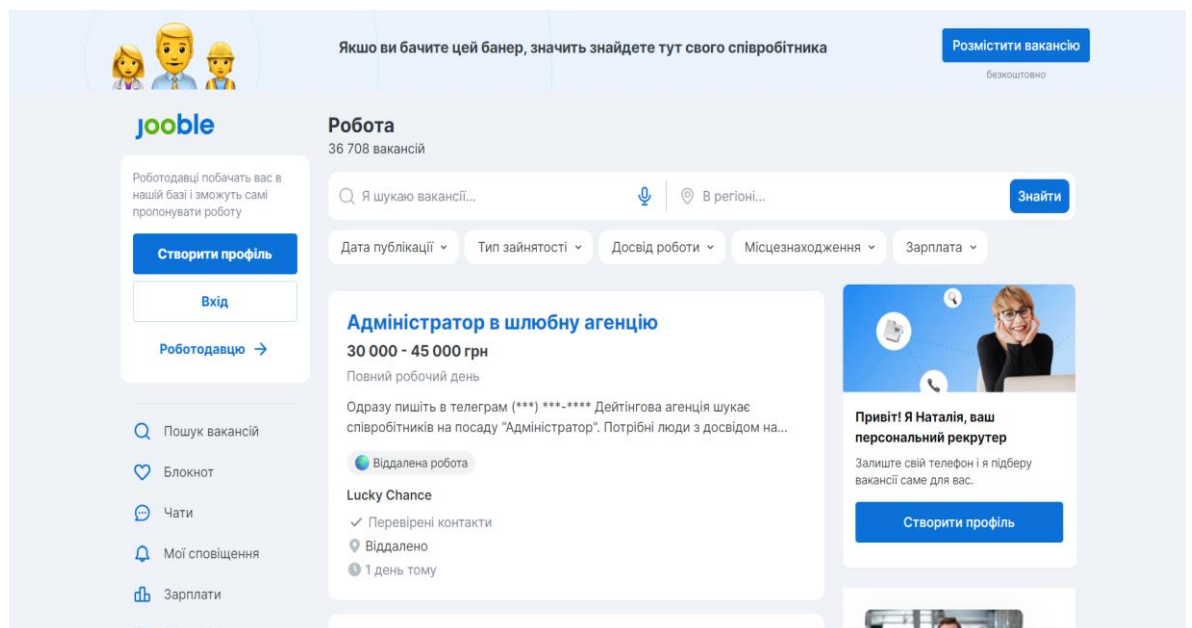


Рис. 1.10 - Сторінка пошуку вакансій сайту Jooble

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 17   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |



## Менше даремних співбесід та витрат на рекрутинг

Кожен третій працівник в Україні шукає роботу на Jooble.

Розмістити вакансію

Безкоштовна реєстрація

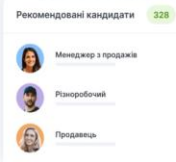
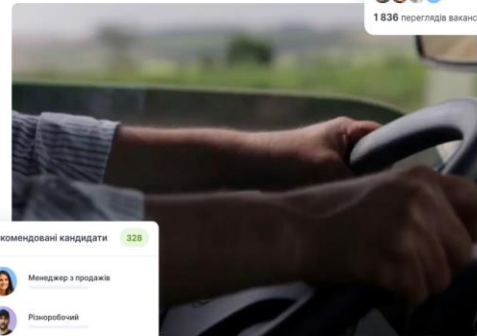


Рис. 1.11 - Головна сторінка сайту Jooble для роботодавців

Рис. 1.12 - Сторінка реєстрації роботодавця сайту Jooble

Після аналізу трьох популярних сайтів для пошуку роботи та працівників (Work.ua, Rabota.ua, та Jooble), можна визначити кілька загальних висновків:

### Інтерфейс та користувацька дружелюбність:

- Work.ua та Rabota.ua вражають своєю простотою, лаконічністю та чіткою структурою. Їхні інтерфейси спрощують навігацію для користувачів.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 18   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

- Jooble, як агрегатор вакансій, також пропонує зручний інтерфейс та широкі можливості фільтрації результатів пошуку.

#### **Активність та оновлення інформації:**

- Work.ua та Rabota.ua відзначаються активною спільнотою користувачів та регулярними оновленнями вакансій, що забезпечує актуальність інформації.
- Jooble вражає своєю масштабністю та географічною доступністю, оскільки працює не тільки на рівні України, а й на міжнародному рівні.

#### **Технологічні рішення:**

- Work.ua та Rabota.ua використовують технологічні нововведення, такі як системи рекомендацій та рейтингів, для покращення ефективності пошуку та вибору роботи чи кандидатів.
- Jooble, як агрегатор, використовує концепцію об'єднання вакансій з різних джерел, що розширює можливості користувачів.

#### **Спрощення пошуку та вибору:**

- Всі три сайти пропонують різноманітні фільтри та опції пошуку, що спрощують завдання користувачів при знаходженні роботи чи працівників.

Аналіз цих аспектів дозволяє визначити кращі практики та стратегії, які можна врахувати при розробці веб-сайту для пошуку роботи та працівників, зокрема, піддавати оновленням та підтримувати активність користувачів, впроваджувати технологічні рішення для покращення ефективності та забезпечення зручності використання.

### **1.3 Вибір архітектури веб-сайту для пошуку роботи та працівників**

Для стабільного та продуктивного функціонування веб-сайту для пошуку роботи та працівників, важливо враховувати вимоги до його архітектури. Архітектура визначає базову організацію системи, компоненти, їх відношення та принципи проектування.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 19   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

Правильно спроектована архітектура сприяє подальшому розвитку, впровадженню, користуванню та модифікації веб-сайту. Використання клієнт-серверної архітектури програмного забезпечення є доцільним для створення подібних систем.

Патерни проектування, як загальні рішення для дизайну архітектури, можуть значно полегшити розробку. Вони не представляють конкретний код, але надають загальні концепції або приклади рішень, які можна налаштовувати під потреби конкретного веб-сайту.

Клієнтська частина веб-сайту відповідає за обробку дій користувача, формування та відправку запитів до серверної частини. Відповіді сервера обробляються та подаються користувачеві. Клієнтська частина взаємодіє з користувачем, відображаючи інформацію про роботу та працівників у зручному та зрозумілому форматі. Використання патернів проектування сприятиме швидкому розвитку веб-сайту з мінімальними витратами завдяки можливості повторного використання коду.

В процесі розробки веб-сайту для пошуку роботи та працівників, вирішено обрати архітектурний підхід, який відповідає сучасним стандартам, забезпечує ефективність та зручність в роботі. У даній роботі буде використано комбінацію PHP-фреймворка Symfony [1] та JavaScript-фреймворка Vue.js [2] для реалізації веб-сайту.

При виборі архітектури, важливо врахувати її сумісність та спроможність роботи разом [3]. Symfony та Vue.js взаємодіють дуже ефективно завдяки використанню API та стандартів обміну даними.

Веб-сайт буде побудований на основі архітектури Model-View-Controller (MVC) [4], де Symfony виступатиме як серверна частина (модель та контролер), а Vue.js буде використовуватися для реалізації фронтенду (вид).

Для оптимізації роботи з ресурсами та забезпечення ефективного управління залежностями використовуватиметься Webpack. Цей інструмент дозволяє збирати, пакувати та оптимізовувати ресурси, такі як JavaScript файли, стилі та інші активи.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 20   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

**Модель-Вид-Контролер (MVC):** Symfony використовує архітектурний патерн MVC для визначення роботи з серверними запитами та базою даних, в той час як Vue.js дозволяє реалізувати MVC на стороні клієнта, що полегшує роботу із змінами інтерфейсу.

**RESTful API [5]:** Symfony вбудовує можливості для створення RESTful API, що ідеально підходить для взаємодії з Vue.js на фронтенді.

**Модель (Symfony) [6]:** Взаємодія з базою даних, обробка бізнес-логіки, та видача даних через RESTful API.

**Вид (Vue.js):** Реалізація інтерфейсу та відображення даних на стороні клієнта.

**Контролер (Symfony) [7]:** Обробка та валідація запитів, взаємодія з моделлю та підготовка даних для відображення.

Також варто зазначити що Symfony фреймворк, використовує різноманітні патерни проектування [8]. Ось кілька патернів, які варті уваги:

**Front Controller:** Symfony використовує Front Controller для централізації обробки запитів.

**Dependency Injection:** Symfony активно використовує патерн Dependency Injection для зручного управління залежностями компонентів та послуг у додатку.

**Observer:** компоненти Symfony використовують патерн Observer для нотифікації слухачів про події, такі як події життєвого циклу додатку.

**Decorator:** Symfony використовує патерн Decorator для розширення функціональності класів без зміни їхньої структури.

**Singleton:** компоненти Symfony використовують патерн Singleton для гарантованого наявності лише одного екземпляра деяких класів.

**Chain of Responsibility:** в Symfony використовують ланцюжки відповідальності для передачі запитів через ланцюжок об'єктів для обробки.

**Strategy:** Symfony може використовувати патерн Strategy для заміни алгоритмів виконання завдань на льоту.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 21   |

**Proxy:** компоненти Symfony використовують патерн Proxy для створення проміжних об'єктів для керування доступом до інших об'єктів.

Ця архітектура дозволить досягти високої продуктивності, зручності у розробці та підтримці, а також забезпечить ефективну взаємодію між серверною та клієнтською частинами веб-сайту для пошуку роботи та працівників.

#### **1.4 Обґрунтування вибору засобів реалізації та вимоги до апаратного забезпечення**

У даному розділі я обґрунтую вибір інструментів та технологій для розробки веб-сайту для пошуку роботи та працівників. Тут буде представлено опис програмних засобів, їх взаємодії та вимоги до апаратного забезпечення для забезпечення ефективної та стабільної роботи системи.

Мною було обрано PHP-фреймворк Symfony для реалізації серверної частини та JavaScript-фреймворк Vue.js для фронтенду. В якості середовища для розробки використовуватиметься PHPStorm [9], який інтегрується із Symfony та Vue.js, сприяючи продуктивності розробки. Крім того, для ефективного управління залежностями та стилями я буду використовувати Webpack [10] та SCSS [11]. В ролі бази даних я буду використовувати MySQL [12] в зв'язці з Beekeeper Studio [13]. А в якості сервера буде використовуватись вбудований сервер Symfony Server [14].

##### **PHPStorm:**

PHPStorm є високопродуктивним інтегрованим середовищем розробки (IDE), спеціально призначеним для роботи з мовою програмування PHP [15]. Вибір PHPStorm базується на кількох ключових перевагах:

- Підтримка PHP: PHPStorm надає широкий спектр інструментів для роботи з PHP, включаючи автодоповнення коду, аналіз коду, відлагодження та інші функції, що підвищують продуктивність розробників.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 22   |

- Інтеграція з Symfony та Vue.js: PHPStorm має покращену підтримку Symfony, що робить розробку серверної частини зручною. Також, інтеграція із Vue.js допомагає в розробці динамічного фронтенду.

- Робота з базами даних: Інтеграція з різними системами керування базами даних (СКБД) спрощує взаємодію з базами даних у процесі розробки.

### **Symfony та Vue.js:**

- Symfony для серверної частини: Symfony є високопродуктивним PHP-фреймворком, який пропонує структурований та розширюваний підхід до розробки веб-додатків. Він забезпечить потрібний рівень абстракції та гнучкості для ефективної реалізації функціональностей веб-сайту. Symfony відповідає сучасним стандартам розробки, забезпечуючи чистий та легко зрозумілий код. Це сприяє підтримці та розширенню проекту в майбутньому. Symfony добре інтегрується з іншими технологіями, що полегшує спільну роботу з базою даних, фронтом та іншими компонентами системи.

- Vue.js для фронтенду: Vue.js є легким та ефективним JavaScript-фреймворком для розробки інтерактивних інтерфейсів. Вибір Vue.js дозволяє швидко та зручно реалізовувати динамічні компоненти та взаємодію на стороні клієнта. Vue.js базується на компонентній структурі, що сприяє чіткому розділенню функціональностей та полегшує управління станом додатку. Vue.js дозволяє ефективно працювати зі стороною клієнта, забезпечуючи високу продуктивність та реактивний підхід до змін стану.

### **Webpack:**

- Webpack для управління залежностями: Webpack дозволяє ефективно управляти та оптимізувати ресурси веб-сайту, включаючи JavaScript [16], CSS та інші файли. Він допомагає покращити продуктивність та швидкість завантаження сторінок.

### **SCSS:**

- SCSS для управління стилями: Використання SCSS (розширений синтаксис таблиць стилів) полегшує розробку та управління стилями. Його

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 23   |

зручні конструкції та можливості сприяють підтримці чистого та організованого коду.

### **MySQL:**

- Надійність та стабільність: MySQL є однією з найпопулярніших відкритих систем керування базами даних, відомою своєю надійністю та стабільністю. Це робить її відмінним вибором для веб-сайтів, які потребують надійного зберігання та доступу до даних.

- Сумісність з Symfony: MySQL добре інтегрується з фреймворком Symfony, що полегшує взаємодію між серверною частиною веб-сайту і базою даних. Це забезпечує ефективність та продуктивність у роботі з даними.

- Забезпечення оптимальної продуктивності: MySQL відомий своєю високою продуктивністю та ефективністю в роботі з обсягами даних, що робить його ідеальним вибором для веб-сайтів, які очікують значного обсягу трафіку та запитів.

- Збереження реляційних даних: MySQL підтримує реляційну модель даних, що сприяє організації та управлінню зв'язками між різними таблицями в базі даних.

### **Beekeeper Studio:**

- Графічний інтерфейс: Beekeeper Studio пропонує зручний графічний інтерфейс для адміністрування та роботи з різними типами баз даних, включаючи MySQL. Його інтуїтивно зрозумілий інтерфейс спрощує взаємодію з базою даних, надаючи можливість створювати та редагувати таблиці, виконувати запити та відстежувати структуру даних.

- Візуалізація запитів: Beekeeper Studio дозволяє візуально складати та оптимізувати SQL-запити для полегшення розробки та оптимізації взаємодії з базою даних. Це забезпечує швидкий та зручний спосіб створення складних запитів.

- Моніторинг та оптимізація продуктивності: Beekeeper Studio має вбудовані інструменти для моніторингу та оптимізації продуктивності бази

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 24   |

даних. Це дозволяє вчасно виявляти та виправляти проблеми ефективності, щоб забезпечити оптимальну продуктивність системи.

### **Symfony Server:**

- Вбудована легкість використання: Symfony надає вбудований сервер, який може бути легко запущений для розробки та тестування. Це дозволяє швидко взяти систему на роботу без необхідності налаштування та управління сторонніми серверами під час розробки.

- Підтримка розгортання: Можливість використовувати внутрішній сервер Symfony також забезпечує легкість у розгортанні системи, оскільки його конфігурація легко переноситься на інші середовища.

Вибір цих інструментів і технологій допомагає створити ефективний та легкозрозумілий код, полегшуючи процес розробки та підтримки веб-сайту для пошуку роботи та працівників.

### **Вимоги до апаратного забезпечення:**

#### **Сервер:**

- Процесор: рекомендується Intel Core i5.
- Оперативна пам'ять (RAM): мінімум 4 ГБ RAM для оптимальної продуктивності.
- Диск: SSD є бажаним для покращення швидкодії завантаження сторінок та обробки запитів.

#### **Мережеве обладнання:**

- Швидкість Інтернет-з'єднання: Високошвидкісне з'єднання з Інтернетом для забезпечення швидкого завантаження сторінок та ефективної роботи із серверами баз даних.

### **Висновки до першого розділу**

У першому розділі було проведено аналіз напрямків використання інформаційних технологій для реалізації веб-сайту пошуку роботи та працівників. Описано проект та поставлено його основні задачі, спрямовані на

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 25   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

полегшення процесу розробки веб сайту для пошуку роботи та працівників в умовах швидко змінюючогося ринку праці.

Аналіз аналогів програмного продукту виявив важливі аспекти та стандарти, що слід враховувати при розробці веб-сайту. Високу планку було встановлено, оцінюючи якісні аналоги та різноманітні підходи до розробки, зокрема на українському ринку.

У розділі розглянуто вибір архітектури веб-сайту, де визначено важливість правильної організації системи та використання патернів проектування. Вибрано комбінацію PHP-фреймворка Symfony та JavaScript-фреймворка Vue.js для забезпечення ефективності та зручності в роботі веб-сайту.

Також розглянуті вибір інструментів та технологій для розробки, такі як PHPStorm, Webpack, SCSS, MySQL, Beekeeper Studio та Symfony Server. Визначено їх взаємодію та вимоги до апаратного забезпечення для забезпечення ефективної та стабільної роботи системи.

В цілому, перший розділ надає чіткий огляд основних аспектів проекту, його мети та завдань, а також обґрунтовує вибір технологічних рішень для успішної реалізації веб-сайту для пошуку роботи та працівників.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 26   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ ДЛЯ ПОШУКУ РОБОТИ ТА ПРАЦІВНИКІВ

### 2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

У цьому пункті буде розглядатись процес проектування веб-сайту, спрямованого на ефективний пошук роботи та працівників. Одним із ключових аспектів є визначення варіантів використання та об'єктно-орієнтованої структури системи. Це важливий етап, що передує розробці та імплементації, оскільки від його результатів залежить подальше функціональне та організаційне вдосконалення веб-сайту.

Також потрібно передбачити різноманітні сценарії використання та розробити об'єктно-орієнтовану структуру системи, що враховує потреби як роботодавців, так і працівників. Головна мета це - забезпечити ефективний та інтуїтивно зрозумілий інтерфейс, сприяючи швидкому та точному пошуку роботи або кваліфікованого персоналу.

#### Бізнес вимоги:

- Основні цілі: створення і розгортання веб-сайту, який допоможе користувачам знайти роботу або найняти працівників шляхом публікації і пошуку вакансій та профілів працівників.

- Представлення проекту: проект буде реалізовано у вигляді веб-сайту, що містить структуровану інформацію про робочі вакансії та працівників, а також функціонал для пошуку та відгуку на них.

#### Вимоги гостя:

- Можливість входу в існуючий профіль
- Можливість використання форми скидання паролю
- Можливість реєстрації профілю працівника
- Можливість реєстрації профілю роботодавця
- Можливість пошуку та фільтрації робочих вакансій

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 27   |

- Можливість перегляду та фільтрацій профілів працівників
- Можливість перегляду сторінок з інформацією про веб-сайт

**Вимоги працівника (рис. 2.1):**

- Можливість редагування та заповнення профілю працівника
- Можливість редагування та заповнення контактів працівника
- Можливість пошуку та фільтрації робочих вакансій
  - Можливість відгукнутись на вакансію
- Можливість перегляду сторінки відгуків на вакансії
  - Можливість видалення відгуку
- Можливість перегляду пропозицій від роботодавців
  - Підтвердження пропозиції
  - Відхилення пропозиції
- Можливість перегляду сторінок з інформацією про веб-сайт

**Вимоги роботодавця (рис. 2.2):**

- Можливість редагування профілю роботодавця
- Можливість редагування інформації про компанію
- Можливість додавати редагувати та видаляти вакансії
- Можливість пошуку та фільтрації працівників
- Можливість перегляду сторінок працівників
- Можливість перегляду пропозицій які були відправлені працівнику
  - Можливість видалення пропозицій
- Можливість перегляду відгуків на вакансії
  - Можливість схвалення відгуку
  - Можливість відхилення відгуку
- Можливість перегляду сторінок з інформацією про веб-сайт

**Вимоги адміністратора (рис. 2.3):**

- Можливість перегляду та редагування користувачів
- Можливість перегляду редагування та додавання нових категорій
- Можливість перегляду та видалення профілів працівників

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 28   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

- Можливість перегляду та видалення контактів працівників
- Можливість перегляду та видалення профілів роботодавців
- Можливість перегляду та видалення інформації про компанію роботодавця
- Можливість перегляду та видалення вакансій
- Можливість перегляду та видалення відгуків на вакансії
- Можливість перегляду та видалення пропозицій працівникам

Після аналізу вимог користувачів веб-сайту можна чітко сформулювати та побудувати діаграму варіантів використання веб-сайту.

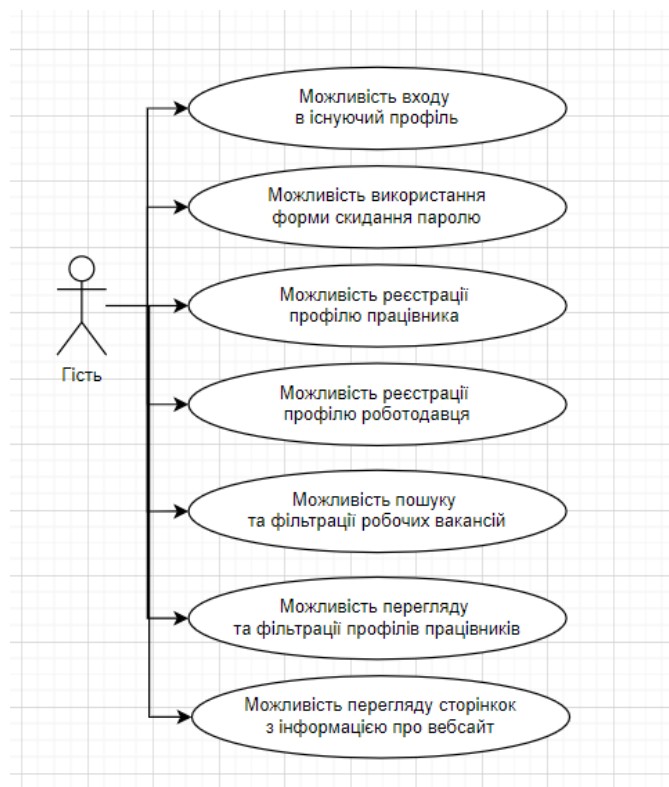


Рис. 2.1 - Діаграма варіантів використання для гостя



Рис. 2.2 - Діаграма варіантів використання для працівника

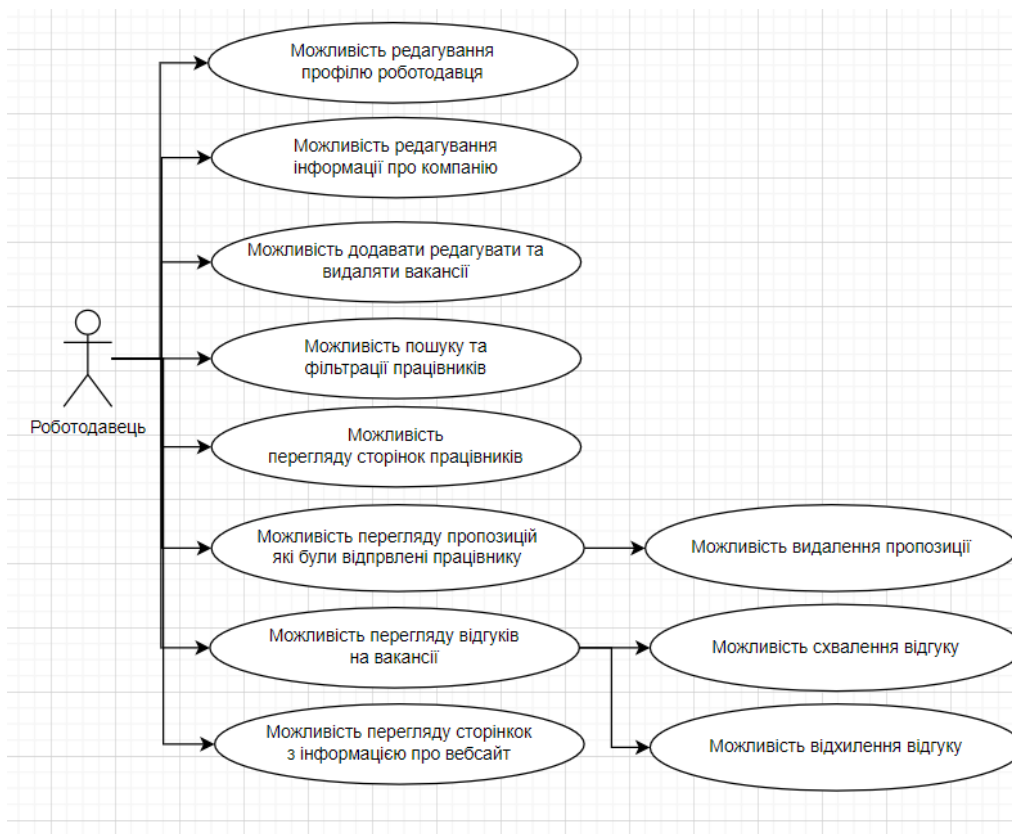


Рис. 2.3 - Діаграма варіантів використання для роботодавця

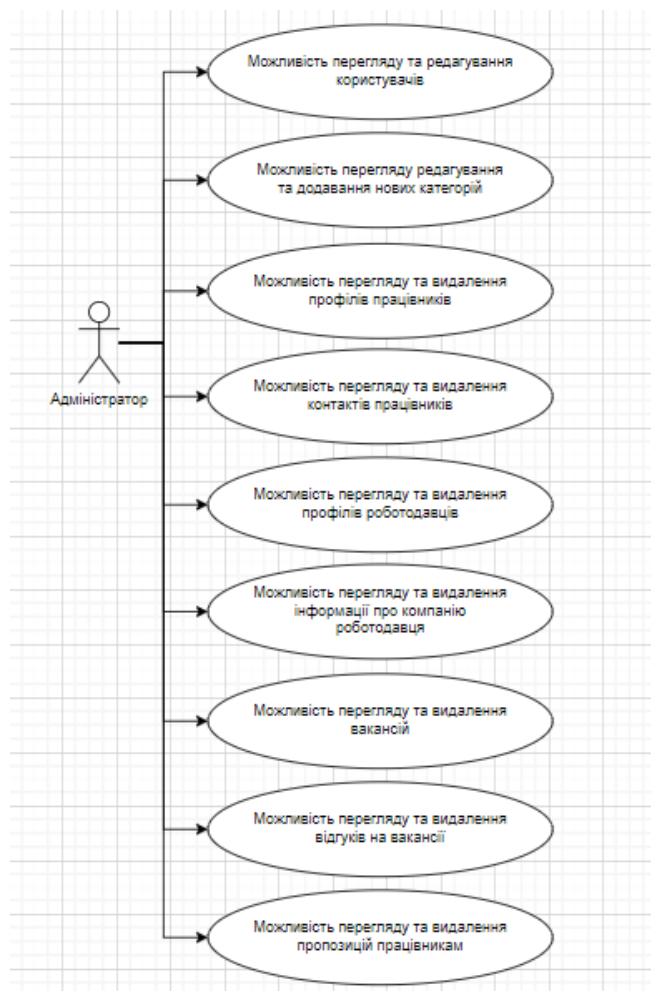


Рис. 2.4 - Діаграма варіантів використання для адміністратора

#### Функціональні вимоги:

- **Реєстрація та авторизація користувачів:** Сайт повинен надавати можливість реєстрації користувачів та авторизації на сайті. Користувачі можуть мати різні ролі, такі як працівник, роботодавець і адміністратор.
- **Можливість додавати та редагувати інформацію:** Працівники повинні мати можливість заповнювати свій робочий профіль та контакти та редагувати інформацію про них. Роботодавці повинні мати можливість заповнювати свій профіль та інформацію про компанію та редагувати їх.
- **Пошук та фільтрація вакансій:** Сайт повинен мати функціонал для пошуку робочих вакансій за різними параметрами, такими як назва

вакансії, місцезнаходження, зарплата тощо. Користувачі повинні мати можливість легко знаходити вакансії, які відповідають їхнім критеріям.

- **Пошук та фільтрація працівників:** Сайт повинен мати функціонал для пошуку працівників за різними параметрами, такими як посада, місцезнаходження, зарплата тощо. Роботодавці повинні мати можливість легко знаходити працівників, які відповідають їхнім критеріям.
- **Відгуки на вакансії:** Працівники повинні мати можливість відгукуватись на вакансії та бачити статус відгуку.
- **Пропозиції працівникам:** Роботодавці повинні мати можливість пропонувати працівникам вакансії які підходять працівникам по їх категорії.
- **Повідомлення:** Сайт повинен надсилати інформативні імейли про дії працівників та роботодавців з відповідними модулями.

#### **Нефункціональні вимоги:**

- **Сприйняття:** Сайт повинен мати інтуїтивно зрозумілий інтерфейс, який дозволить звичайним користувачам вивчити його протягом 10-15 хвилин. Час відгуку на типові запити повинен бути не більше 5 секунд. Інтерфейс повинен бути зрозумілим і містити відповідні підказки для користувачів.
- **Надійність:** Сайт повинен бути доступним протягом 24 годин на добу, 7 днів на тиждень. Середній час безвідмовної роботи повинен становити не менше 30 днів. Максимальна норма помилок та дефектів в роботі системи - 1 помилка на тисячу рядків коду.
- **Продуктивність:** Система повинна підтримувати мінімум 1000 одночасно працюючих користувачів, щоб забезпечити плавну роботу сайту.
- **Можливість експлуатації:** Система повинна мати можливість масштабування, щоб збільшити її продуктивність зі зростанням користувачів без впливу на її роботу.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 32   |

Після аналізу функціональних вимог можна перейти до створення діаграми класів (рис. 2.5), яка буде відображати основні класи системи, їх атрибути, методи та взаємозв'язки.

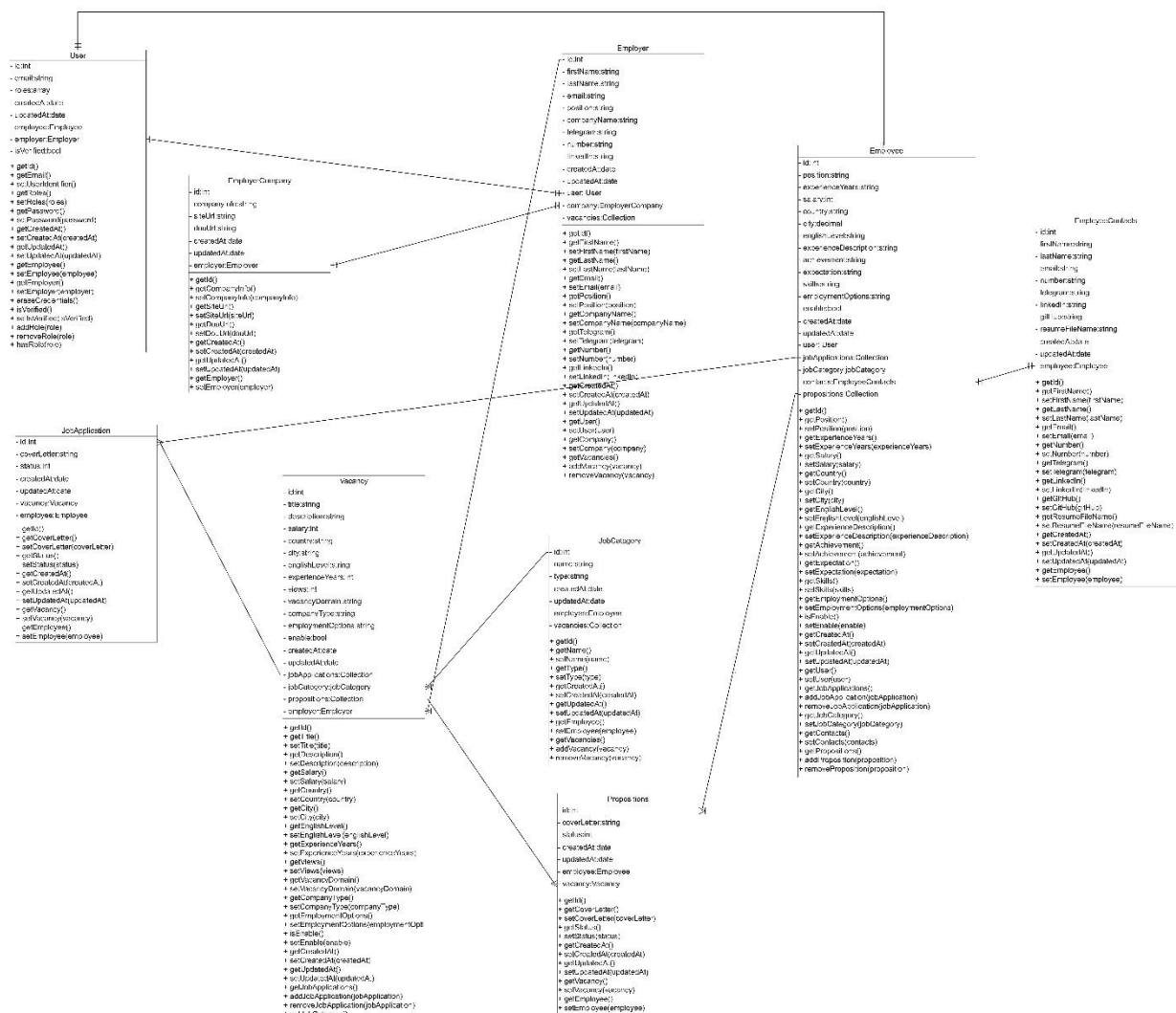


Рис. 2.5 - Діаграма класів

## Опис класів які відображені на діаграмі:

**Клас “Vacancy” (Вакансія)** - Цей клас представляє вакансію, створену роботодавцем. Вакансія містить інформацію про посаду, опис роботи, вимоги до кандидата, рівень зарплати, місцезнаходження, необхідний рівень володіння англійською мовою, необхідний досвід роботи та інші деталі. Вакансія належить до певної категорії роботи (JobCategory) і може мати багато відгуків

(JobApplication) та пропозицій цієї вакансії для працівників (Propositions). Вона також пов'язана з роботодавцем (Employer), який її створив.

**Клас “JobCategory” (Категорія Роботи)** - Цей клас представляє категорію роботи, до якої можуть належати вакансії. Категорія роботи має назву і тип, та містить список вакансій, що належать до неї. Крім того, категорія роботи може бути пов'язана з багатьма працівниками (Employee), які працюють у цій категорії.

**Клас Propositions (Пропозиції)** - Цей клас представляє пропозиції роботи, які можуть бути відправлені працівникам роботодавцями. Пропозиція містить супровідний лист, статус пропозиції, а також дати створення та оновлення. Пропозиція пов'язана з певною вакансією (Vacancy) та працівником (Employee).

**Клас JobApplication (Заявка на Роботу)** - Цей клас представляє заявку на роботу, подану працівником на конкретну вакансію. Заявка містить супровідний лист, статус заявки, а також дати створення та оновлення. Заявка пов'язана з певною вакансією (Vacancy) та працівником (Employee).

**Клас Employer (Роботодавець)** - Цей клас представляє роботодавця, який може створювати вакансії та керувати ними. Роботодавець містить інформацію про ім'я, прізвище, електронну адресу, посаду, назву компанії та контактні дані. Він пов'язаний з користувачем (User), який може входити в систему, та з компанією (EmployerCompany), до якої він належить. Роботодавець також має список вакансій (Vacancy), які він створив.

**Клас Employee (Працівник)** - Цей клас представляє працівника, який може подавати заявки на роботу та отримувати пропозиції роботи. Працівник містить інформацію про посаду, досвід роботи, зарплату, місцезнаходження, рівень володіння англійською мовою, опис досвіду роботи, досягнення, очікування, навички та інші деталі. Він пов'язаний з користувачем (User), який може входити в систему, та з категорією роботи (JobCategory), в якій він працює. Працівник також має контакти (EmployeeContacts), список заявок на роботу (JobApplication) та пропозицій роботи (Propositions).

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 34   |

**Клас User (Користувач)** - Цей клас представляє користувача системи, який може бути працівником або роботодавцем. Користувач містить інформацію про електронну адресу, ролі (наприклад, працівник, роботодавець, адміністратор), пароль, а також дати створення та оновлення. Користувач може бути пов'язаний з працівником (Employee) або роботодавцем (Employer) і містить інформацію про верифікацію.

**Клас EmployerCompany (Компанія Роботодавця)** - Цей клас представляє компанію роботодавця, до якої може належати роботодавець. Компанія містить інформацію про компанію, веб-сайт, профіль на Dou.ua, а також дати створення та оновлення. Вона пов'язана з роботодавцем (Employer), який є її представником.

**Клас EmployeeContacts (Контакти Працівника)** - Цей клас представляє контакти працівника, які можуть включати різні засоби зв'язку, такі як телефонний номер, електронна адреса, посилання на профілі в соціальних мережах (LinkedIn, GitHub) та резюме. Клас також містить інформацію про ім'я, прізвище, а також дати створення та оновлення. Контакти пов'язані з працівником (Employee).

## 2.2 Розробка бази даних системи

Для збереження інформації веб-сайту я використав базу даних MySQL. MySQL - це популярна реляційна система керування базами даних (RDBMS), що працює на основі SQL (Structured Query Language). MySQL є відкритим програмним забезпеченням, що забезпечує високу продуктивність, надійність та легкість у використанні. Вона широко використовується у веб-розробці завдяки своїй здатності ефективно обробляти великі обсяги даних і підтримувати масштабованість.

**До бази даних було вирішено додати дев'ять таблиць:**

1. user – таблиця user зберігає основну інформацію про користувачів, включаючи їхні електронні адреси, паролі, ролі, а також інформацію про

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 35   |

те, чи є користувач підтвердив свій імейл. Користувач може бути як працівником, так і роботодавцем, що відображається через відповідні зв'язки з таблицями employee та employer.

Таблиця 2.1

| Назва       | Тип даних    | ПК | ЗК | Опис поля                            |
|-------------|--------------|----|----|--------------------------------------|
| id          | int          | +  | -  | Унікальний ідентифікатор користувача |
| email       | varchar(180) | -  | -  | Електронна адреса користувача        |
| roles       | json         | -  | -  | Ролі користувача                     |
| password    | varchar(255) | -  | -  | Пароль користувача                   |
| created_at  | datetime     | -  | -  | Дата створення запису                |
| updated_at  | datetime     | -  | -  | Дата останнього оновлення запису     |
| is_verified | tinyint(1)   | -  | -  | Верифікація користувача              |

2. employee – таблиця employee містить детальну інформацію про працівників, включаючи їхню посаду, досвід роботи, зарплатні очікування, рівень англійської мови, навички та інші важливі дані, що допомагають роботодавцям знайти відповідних кандидатів.

Таблиця 2.2

| Назва                  | Тип даних    | ПК | ЗК | Опис поля                           |
|------------------------|--------------|----|----|-------------------------------------|
| id                     | int          | +  | -  | Унікальний ідентифікатор працівника |
| user_id                | int          | -  | +  | Ідентифікатор користувача           |
| job_category_id        | int          | -  | +  | Ідентифікатор категорії             |
| position               | varchar(255) | -  | -  | Посада працівника                   |
| experience_years       | int          | -  | -  | Кількість років досвіду             |
| salary                 | int          | -  | -  | Зарплатні очікування                |
| country                | varchar(255) | -  | -  | Країна проживання                   |
| city                   | varchar(255) | -  | -  | Місто проживання                    |
| english_level          | varchar(255) | -  | -  | Рівень англійської мови             |
| skills                 | longtext     | -  | -  | Навички                             |
| experience_description | longtext     | -  | -  | Опис досвіду                        |

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 36   |

| Назва              | Тип даних    | ПК | ЗК | Опис поля                        |
|--------------------|--------------|----|----|----------------------------------|
| achievement        | longtext     | -  | -  | Опис досягнень                   |
| expectation        | longtext     | -  | -  | Опис очікувань від роботи        |
| employment_options | varchar(255) | -  | -  | Тип зайнятості                   |
| enable             | tinyint(1)   | -  | -  | Статус акаунту працівника        |
| created_at         | datetime     | -  | -  | Дата створення запису            |
| updated_at         | datetime     | -  | -  | Дата останнього оновлення запису |

3. employee\_contacts – таблиця employee\_contacts зберігає контактну інформацію працівників, таку як ім'я, прізвище, номер телефону, електронна пошта, посилання на профілі в соціальних мережах, що дозволяє роботодавцям легко зв'язатися з працівниками.

Таблиця 2.3

| Назва       | Тип даних    | ПК | ЗК | Опис поля                                      |
|-------------|--------------|----|----|------------------------------------------------|
| id          | int          | +  | -  | Унікальний ідентифікатор контактної інформації |
| first_name  | varchar(255) | -  | -  | Ім'я працівника                                |
| last_name   | varchar(255) | -  | -  | Прізвище працівника                            |
| email       | varchar(255) | -  | -  | Електронна адреса                              |
| number      | varchar(255) | -  | -  | Номер телефону                                 |
| telegram    | varchar(255) | -  | -  | Посилання на Telegram                          |
| linked_in   | varchar(255) | -  | -  | Посилання на профіль у LinkedIn                |
| git_hub     | varchar(255) | -  | -  | Посилання на профіль у GitHub                  |
| created_at  | datetime     | -  | -  | Дата створення запису                          |
| updated_at  | datetime     | -  | -  | Дата останнього оновлення запису               |
| employee_id | int          | -  | +  | Ідентифікатор працівника                       |

4. employer – Таблиця employer містить інформацію про роботодавців, включаючи їхні імена, контактні дані та посади в компанії. Ця таблиця зв'язана з таблицею employer\_companу, яка містить інформацію про компанії, в яких працюють роботодавці.

Таблиця 2.4

| Назва        | Тип даних    | ПК | ЗК | Опис поля                            |
|--------------|--------------|----|----|--------------------------------------|
| id           | int          | +  | -  | Унікальний ідентифікатор роботодавця |
| first_name   | varchar(255) | -  | -  | Ім'я роботодавця                     |
| last_name    | varchar(255) | -  | -  | Прізвище роботодавця                 |
| position     | varchar(255) | -  | -  | Посада роботодавця                   |
| email        | varchar(255) | -  | -  | Електронна адреса                    |
| company_name | varchar(255) | -  | -  | Назва компанії                       |
| telegram     | varchar(255) | -  | -  | Посилання на Telegram                |
| number       | varchar(255) | -  | -  | Номер телефону                       |
| linked_in    | varchar(255) | -  | -  | Посилання на профіль у LinkedIn      |
| created_at   | datetime     | -  | -  | Дата створення запису                |
| updated_at   | datetime     | -  | -  | Дата останнього оновлення запису     |
| user_id      | int          | -  | +  | Ідентифікатор користувача            |

5. employer\_company – таблиця employer\_company зберігає інформацію про компанії, включаючи назву компанії, інформацію про компанію, веб-сайт та дату заснування. Ця інформація допомагає працівникам дізнатися більше про потенційного роботодавця.

Таблиця 2.5

| Назва        | Тип даних    | ПК | ЗК | Опис поля                         |
|--------------|--------------|----|----|-----------------------------------|
| id           | int          | +  | -  | Унікальний ідентифікатор компанії |
| employer_id  | int          | -  | +  | Ідентифікатор роботодавця         |
| company_info | longtext     | -  | -  | Інформація про компанію           |
| site_url     | varchar(255) | -  | -  | Веб-сайт компанії                 |
| dou_url      | varchar(255) | -  | -  | Посилання на профіль на Dou.ua    |
| created_at   | datetime     | -  | -  | Дата створення запису             |
| updated_at   | datetime     | -  | -  | Дата останнього оновлення запису  |

6. job\_application – таблиця job\_application містить дані про відгуки на вакансії включаючи супровідні листи, статус заявок, дати створення та

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 38   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

оновлення, а також посилання на відповідні вакансії та працівників, які подали ці відгуки.

Таблиця 2.6

| Назва        | Тип даних    | ПК | ЗК | Опис поля                        |
|--------------|--------------|----|----|----------------------------------|
| id           | int          | +  | -  | Унікальний ідентифікатор відгуку |
| cover_letter | longtext     | -  | -  | Супровідний лист                 |
| status       | varchar(255) | -  | -  | Статус заявки                    |
| created_at   | datetime     | -  | -  | Дата створення запису            |
| updated_at   | datetime     | -  | -  | Дата останнього оновлення запису |
| vacancy_id   | int          | -  | +  | Ідентифікатор вакансії           |
| employee_id  | int          | -  | +  | Ідентифікатор працівника         |

7. job\_category – таблиця job\_category зберігає категорії вакансій, такі як ІТ, маркетинг, продажі тощо. Кожна категорія може містити багато вакансій, що дозволяє класифікувати робочі місця за певними категоріями для зручного пошуку.

Таблиця 2.7

| Назва      | Тип даних    | ПК | ЗК | Опис поля                          |
|------------|--------------|----|----|------------------------------------|
| id         | int          | +  | -  | Унікальний ідентифікатор категорії |
| name       | varchar(255) | -  | -  | Назва категорії                    |
| type       | varchar(255) | -  | -  | Тип категорії                      |
| created_at | datetime     | -  | -  | Дата створення запису              |
| updated_at | datetime     | -  | -  | Дата останнього оновлення запису   |

8. propositions – таблиця propositions містить пропозиції від роботодавців, включаючи супровідні листи, статус пропозицій, дати створення та оновлення, а також посилання на відповідні вакансії та працівників, яким ці пропозиції адресовані.

Таблиця 2.8

| Назва | Тип даних | ПК | ЗК | Опис поля                           |
|-------|-----------|----|----|-------------------------------------|
| id    | int       | +  | -  | Унікальний ідентифікатор пропозиції |

| Назва        | Тип даних    | ПК | ЗК | Опис поля                        |
|--------------|--------------|----|----|----------------------------------|
| cover_letter | longtext     | -  | -  | Супровідний лист                 |
| status       | varchar(255) | -  | -  | Статус пропозиції                |
| created_at   | datetime     | -  | -  | Дата створення запису            |
| updated_at   | datetime     | -  | -  | Дата останнього оновлення запису |
| vacancy_id   | int          | -  | +  | Ідентифікатор вакансії           |
| employee_id  | int          | -  | +  | Ідентифікатор працівника         |

9. vacancy – таблиця vacancy зберігає інформацію про вакансії, включаючи назву посади, опис, зарплатні очікування, досвід роботи, необхідні навички та інші вимоги до кандидатів. Вакансії можуть бути класифіковані за категоріями, що зберігаються у таблиці job\_category, і зв'язані з роботодавцями, що зберігаються у таблиці employer.

Таблиця 2.9

| Назва            | Тип даних    | ПК | ЗК | Опис поля                         |
|------------------|--------------|----|----|-----------------------------------|
| id               | int          | +  | -  | Унікальний ідентифікатор вакансії |
| job_category_id  | int          | -  | +  | Ідентифікатор категорії вакансії  |
| title            | varchar(255) | -  | -  | Назва вакансії                    |
| description      | longtext     | -  | -  | Опис вакансії                     |
| salary           | int          | -  | -  | Зарплатня                         |
| city             | varchar(255) | -  | -  | Місто                             |
| country          | varchar(255) | -  | -  | Країна                            |
| experience_years | int          | -  | -  | Вимоги до досвіду                 |
| english_level    | varchar(255) | -  | -  | Вимоги до рівня англійської мови  |
| views            | int          | -  | -  | Кількість переглядів вакансії     |
| vacancy_domain   | varchar(255) | -  | -  | Домен вакансії                    |
| company_type     | varchar(255) | -  | -  | Тип компанії                      |
| employer_id      | int          | -  | +  | Ідентифікатор роботодавця         |
| enable           | tinyint(1)   | -  | -  | Статус публікації вакансії        |

| Назва              | Тип даних    | ПК | ЗК | Опис поля                        |
|--------------------|--------------|----|----|----------------------------------|
| employment_options | varchar(255) | -  | -  | Тип зайнятості                   |
| created_at         | datetime     | -  | -  | Дата створення запису            |
| updated_at         | datetime     | -  | -  | Дата останнього оновлення запису |

Ця структура бази даних забезпечує зручну організацію і доступ до даних, необхідних для функціонування веб-сайту для пошуку роботи та працівників

## 2.3 Проектування та реалізація алгоритмів роботи системи

Веб-сайту для пошуку роботи та працівників вимагає ретельного планування та розробки алгоритмів, що забезпечують ефективну взаємодію між працівниками, роботодавцями та системою в цілому. Основна мета це створити зручну та функціональну платформу, яка допомагає працівникам знайти відповідні вакансії, а роботодавцям - знайти кваліфікованих працівників.

У цьому розділі я опишу загальний алгоритми роботи системи, що продемонструє основні етапи взаємодії користувачів із веб-сайтом, починаючи від реєстрації та створення профілю, до пошуку вакансій, подачі заявок та отримання пропозицій від роботодавців. Описуються ключові елементи системи та їхня взаємодія під час роботи програмного комплексу, що забезпечує надійність, ефективність та зручність використання веб-сайту.

### Алгоритм роботи веб-сайту для неавторизованого користувача:

- після переходу на сайт користувач бачить головну сторінку
- з головної сторінки користувач може перейти на сторінку входу, реєстрації, списку працівників та вакансій
- на сторінці входу користувач має можливість увійти в акаунт або скористатись можливістю скинути пароль
- на сторінці реєстрації користувач може створити акаунт і обрати тип акаунту працівник, або роботодавець

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 41   |

– на сторінці вакансій користувач може переглядати усі активні вакансії та фільтрувати їх за бажанням, але для відгуку на конкретну вакансію йому потрібно буде зареєструватись

– на сторінці працівників користувачу будуть доступні всі активні працівники і їх профілі

**Алгоритм роботи веб-сайту для працівника:**

- після авторизації працівник бачить головну сторінку зі списком вакансій
- працівник може перейти в особистий профіль для редагування інформації та своїх контактів
- на сторінці вакансій працівник може переходити на сторінку конкретної вакансії, та відправляти відгук на неї
- на сторінці відгуків працівник може бачити статус відгуку та видалити відгук, якщо роботодавець ще не виконав з ним ніяких дій
- на сторінці пропозицій працівник бачить список пропозицій від роботодавців і він може або прийняти пропозицію, або відхилити

**Алгоритм роботи веб-сайту роботодавця:**

- після авторизації роботодавець бачить головну сторінку зі списком працівників
- роботодавець може відкрити свій профіль та відредагувати інформацію про себе та компанію
- на сторінці працівників роботодавець може переглядати публічні профілі працівників та пропонувати їм вакансії
- на сторінці відгуків роботодавець бачить список відгуків на його вакансії де він може схвалити відгук або відхилити
- на сторінці пропозицій роботодавець може переглядати пропозиції які він надсилав працівники, і якщо працівник ще не виконав ніяких дій пропозицію можна видалити

**Алгоритм роботи веб-сайту для адміністратора:**

- адміністратор може перейти в адмін панель натиснувши відповідну кнопку

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 42   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

- на головній сторінці у адміністратора буде вибір сторінок
- на сторінці користувачів адміністратор зможе переглядати користувачів веб сайту редагувати та видаляти їх
- на сторінці категорій можна додавати редагувати та видаляти категорії для вакансій та працівників
- на сторінці працівників можна переглядати та видаляти інформацію про профілі працівників
- на сторінці контактів працівників адміністратор може переглядати та видаляти контакти працівників
- на сторінці роботодавців адміністратор може переглядати та видаляти інформацію про роботодавців
- на сторінці компаній роботодавців адміністратор може переглядати та видаляти інформацію про компанії
- на сторінці вакансій адміністратор може переглядати наявні вакансії і видаляти їх
- на сторінці відгуків на вакансії адміністратор може переглядати та видаляти відгуки
- на сторінці пропозицій адміністратор може переглядати та видаляти пропозиції

На базі цих алгоритмів можна побудувати діаграми дій основного функціоналу веб-сайту:

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 43   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

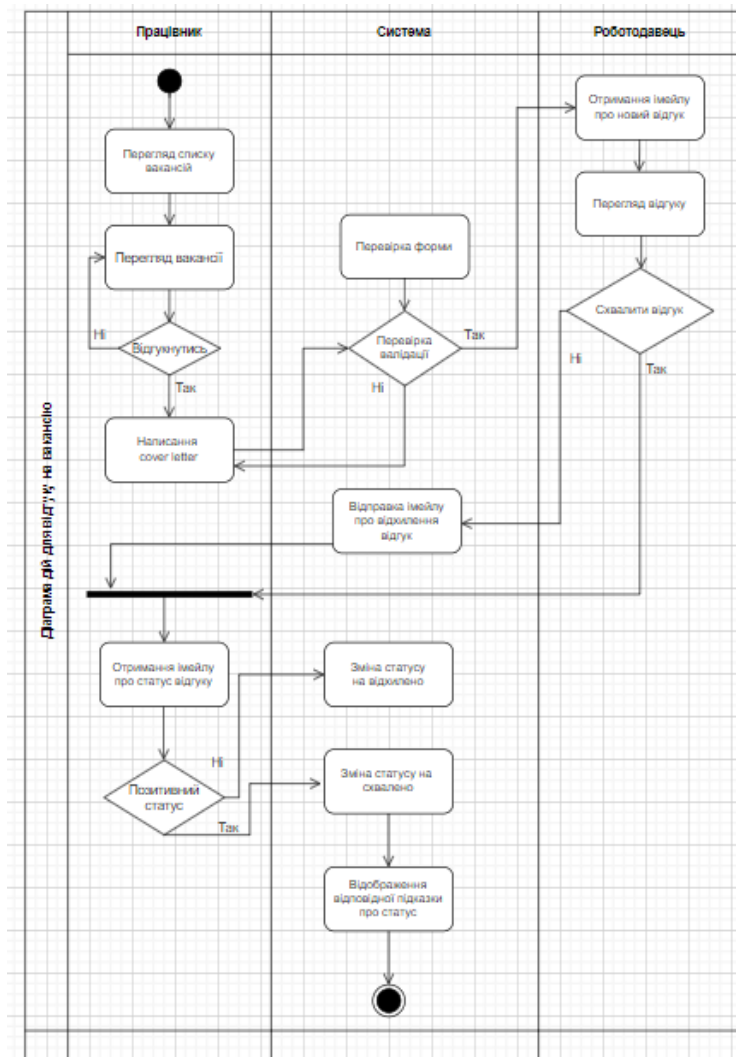


Рис. 2.6 - Діаграма дій для відгуку на вакансію

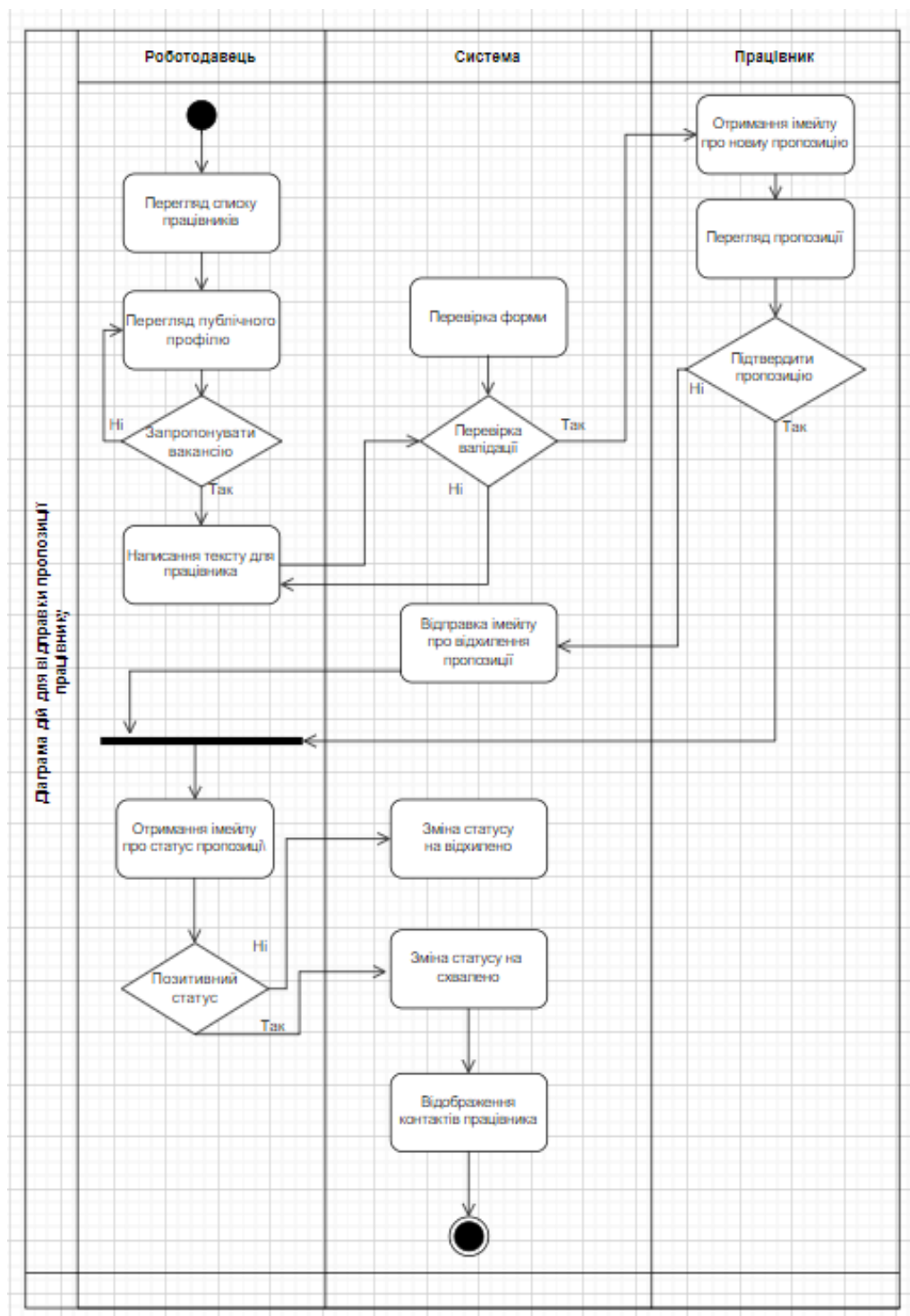


Рис. 2.7 - Діаграма дій для відправки пропозиції працівнику

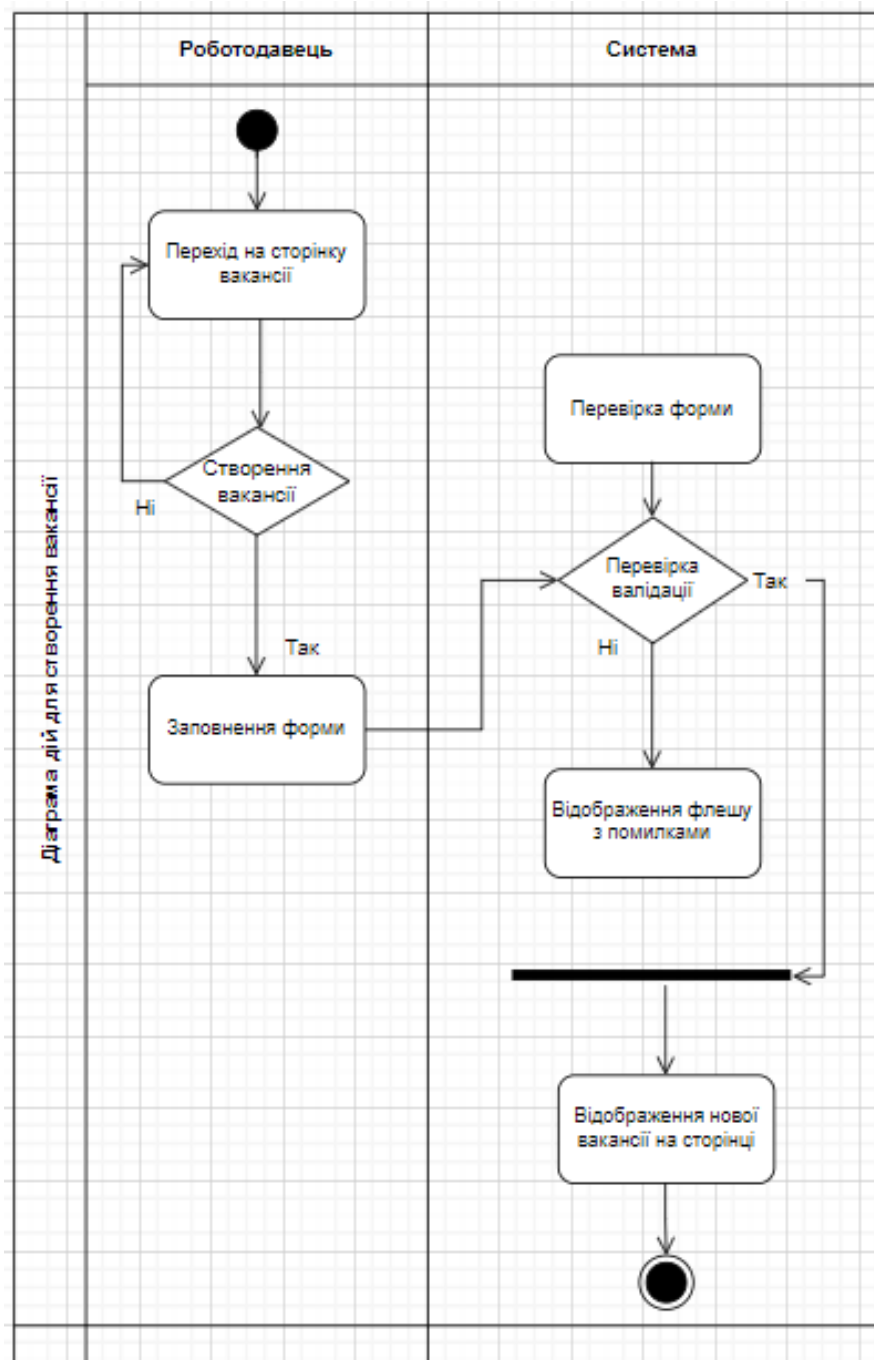


Рис. 2.8 - Діаграма дій для створення вакансії

## 2.4 Реалізація функціоналу веб-сайту для пошуку роботи

У цьому розділі ми детально розглянемо процес реалізації функціоналу веб-сайту для пошуку роботи. Метою даного веб-сайту є забезпечення ефективної взаємодії між роботодавцями та потенційними працівниками, що дозволяє швидко знаходити відповідні вакансії та кандидатів.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 46   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

Розробка даного програмного комплексу включає кілька ключових етапів, серед яких проектування архітектури системи, створення бази даних, розробка серверної частини (Symfony) та клієнтської частини (Vue.js). Я представлю фрагменти коду, які є найбільш складними з точки зору реалізації, і потребують додаткового пояснення для кращого розуміння їх функціональності.

### Розглянемо основні етапи реєстрації нового користувача:

- Функція `registrationAction` на бекенді відповідає за реєстрацію користувача вона приймає дані з реквесту і передає їх в Symfony Form [17] для їх обробки, і при успішній реєстрації надсилає лист для підтвердження імейлу [18].

Фрагмент коду:

```
$user = new User();
$requestData = json_decode($request->getContent(), true);
$accountType = $requestData['accountType'];
$form = $this->createForm(RegistrationFormType::class, $user);
$form->submit($requestData);

if ($form->isSubmitted() && $form->isValid() && $accountType) {
    $user->setPassword(
        $userPasswordHasher->hashPassword(
            $user,
            $form->get('password')->getData()
        )
    );

    if ($accountType === 'employee') {
        $user->addRole('ROLE_EMPLOYEE');
        $employee = new Employee();
        $employee->setUser($user);
        $this->entityManager->persist($employee);
        $employeeContacts = new EmployeeContacts();
        $employeeContacts->setEmployee($employee);
        $employeeContacts->setEmail($requestData['email']);
        $this->entityManager->persist($employeeContacts);
    } elseif ($accountType === 'employer') {
        $user->addRole('ROLE_EMPLOYER');
        $employer = new Employer();
        $employer->setUser($user);
        $employer->setEmail($requestData['email']);
        $this->entityManager->persist($employer);
        $employerCompany = new EmployerCompany();
        $employerCompany->setEmployer($employer);
        $this->entityManager->persist($employerCompany);
    }

    $this->entityManager->persist($user);
    $this->entityManager->flush();
    $this->emailVerifier->sendEmailConfirmation($user);
}
```

Рис. 2.9 - Фрагмент функції `registrationAction`

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 47   |

- Передавання даних з фронтенду для реєстрації

Фрагмент коду:

```
<div class="registration_form">
  <div class="registration_input-group">
    <input type="email" placeholder="Email" v-model="user.email">
  </div>
  <div class="registration_input-group">
    <input :type="passwordFieldType" placeholder="Пароль" v-
model="user.password">
    <i class="fa fa-eye" v-if="passwordFieldType === 'text'"
@click="togglePasswordVisibility"></i>
    <i class="fa fa-eye-slash" v-else
@click="togglePasswordVisibility"></i>
  </div>
  <h1 class="registration_accountType">Оберіть тип акаунту</h1>
  <div class="registration_input-group">
    <select v-model="user.accountType">
      <option value="employee">Працівник</option>
      <option value="employer">Роботодавець</option>
    </select>
  </div>
  <div class="registration_button-container">
    <button @click="submitForm">Зареєструватись</button>
  </div>
```

Рис. 2.10 - Фрагмент форми реєстрації

```
submitForm() {
  axios.post('/app/register', this.user)
    .then(() => {
      myNotifyFlash("Реєстрація успішна, тепер ви можете увійти.",
'success', true);
      window.location.href = '/login';
    }).catch(error => {
      if (error.response && error.response.data &&
error.response.data.errors) {
        const errors = error.response.data.errors;
        for (const key in errors) {
          if (errors.hasOwnProperty(key)) {
            myNotifyFlash(errors[key], 'error');
          }
        }
      }
    })
}
```

Рис. 2.11 - Функція сабміту форми реєстрації

Розглянемо клас `HttpErrorListener` який відповідає за відстеження помилок для яких потрібно рендерити [19] відповідні сторінки:

Фрагмент коду:

```
public function onKernelException(ExceptionEvent $event): void
{
    $exception = $event->getThrowable();

    if ($exception instanceof NotFoundHttpException) {
        $response = new Response($this->templating->
```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 48   |

```

>render('Exceptions/404page.html.twig', [
    'title' => 'dwork | Помилка 404',
]);
$event->setResponse($response);
} elseif ($exception instanceof AccessDeniedHttpException) {
    $response = new Response($this->templating-
>render('Exceptions/403page.html.twig', [
    'title' => 'dwork | Помилка 403',
]);
$event->setResponse($response);
} elseif ($exception instanceof HttpException) {
    $response = new Response($this->templating-
>render('Exceptions/401page.html.twig', [
    'title' => 'dwork | Помилка 401',
]);
$event->setResponse($response);
}
}
}

```

Рис. 2.12 - Функція onKernelException

Розглянемо функціонал відгуку на вакансію:

- Функція addVacancyAction на бекенді відповідає за створення нової вакансії вона приймає дані з реквесту і передає їх в Symfony Form для їх обробки.

Фрагмент коду:

```

$vacancy = new Vacancy();
$user = $this->getUser();
$employer = $user->getEmployer();
$requestData = json_decode($request->getContent(), true);
$form = $this->createForm(VacancyType::class, $vacancy);
$form->submit($requestData);

if ($form->isSubmitted() && $form->isValid()) {
    if ($requestData['jobCategory'] === null) {
        $vacancy->setJobCategory(null);
    } else {
        $jobCategory = $this->entityManager-
>getRepository(JobCategory::class)->findOneBy(['id' =>
$requestData['jobCategory']['id']]);
        $vacancy->setJobCategory($jobCategory);
    }

    $vacancy->setEmployer($employer);
    $this->entityManager->persist($vacancy);
    $this->entityManager->flush();

    return new JsonResponse($vacancy, Response::HTTP_CREATED);
}

```

Рис. 2.13 - Фрагмент функції addVacancyAction

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 49   |

- Метод createVacancy який відправляє дані на бекенд

Фрагмент коду:

```
createVacancy() {
  if (this.vacancy.jobCategory === null) {
    myNotifyFlash('Помилка: категорія не може бути порожньою.',
    'error', false);
  }
  axios.post('/app/vacancy', this.vacancy)
    .then(() => {
      myNotifyFlash('Вакансію успішно створено.', 'success', true);
      window.location.href =
      `/vacancies/employer/${this.employerId}`;
    })
    .catch(error => {
      if (error.response && error.response.data &&
      error.response.data.errors) {
        const errors = error.response.data.errors;
        for (const key in errors) {
          if (errors.hasOwnProperty(key)) {
            myNotifyFlash(errors[key], 'error');
            this.scrollToTop();
          }
        }
      }
    });
},
```

Рис. 2.14 - Метод createVacancy

## Висновки до другого розділу

У другому розділі було представлено комплексний процес проектування та розробки веб-сайту для пошуку роботи та працівників. На початку розділу ми розглянули варіанти використання та об'єктно-орієнтовану структуру системи, що дозволило сформулювати основні сценарії взаємодії користувачів із системою. Цей етап є критичним для забезпечення інтуїтивного та ефективного користувацького досвіду.

Наступним кроком було розроблення бази даних системи з використанням MySQL, популярної реляційної системи керування базами даних. Детально описано структуру таблиць, їхні атрибути та зв'язки, що забезпечує надійне та ефективне збереження і обробку даних.

Проектування та реалізація алгоритмів роботи системи було зосереджено на створенні функціональних та надійних процесів, що забезпечують

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 50   |

безперебійну взаємодію між різними компонентами системи. Описано загальні алгоритми роботи системи, які демонструють основні етапи взаємодії користувачів із веб-сайтом, від реєстрації до отримання пропозицій від роботодавців.

Завершальний етап розглянув детальну реалізацію деяких функцій веб-сайту. Ми зосередились на ключових аспектах розробки серверної частини за допомогою Symfony та клієнтської частини за допомогою Vue.js. Було представлено фрагменти коду, які є найбільш складними з точки зору реалізації, з додатковими поясненнями для кращого розуміння їх функціональності.

Загалом, другий розділ надав комплексний огляд проектування, розробки та реалізації веб-сайту для пошуку роботи та працівників, підкресливши важливість кожного етапу в створенні ефективної та надійної платформи.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 51   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-САЙТОМ ДЛЯ ПОШУКУ РОБОТИ ТА ПРАЦІВНИКІВ

### 3.1 Порядок встановлення та налаштування параметрів системи

Встановлення та налаштування платформи передбачає розгортання всіх її компонентів та конфігурацію як окремих елементів, так і системи в цілому. Цей процес включає детальне налаштування кожного модуля, забезпечуючи їхню коректну взаємодію та інтеграцію. Для детального опису архітектури платформи було створено діаграму розгортання, яка наочно демонструє взаємозв'язки між різними компонентами системи та їхнє розміщення у середовищі розгортання. Діаграма допомагає зрозуміти, як саме налаштування впливають на функціонування веб-сайту, що забезпечує ефективність і стабільність всієї платформи.

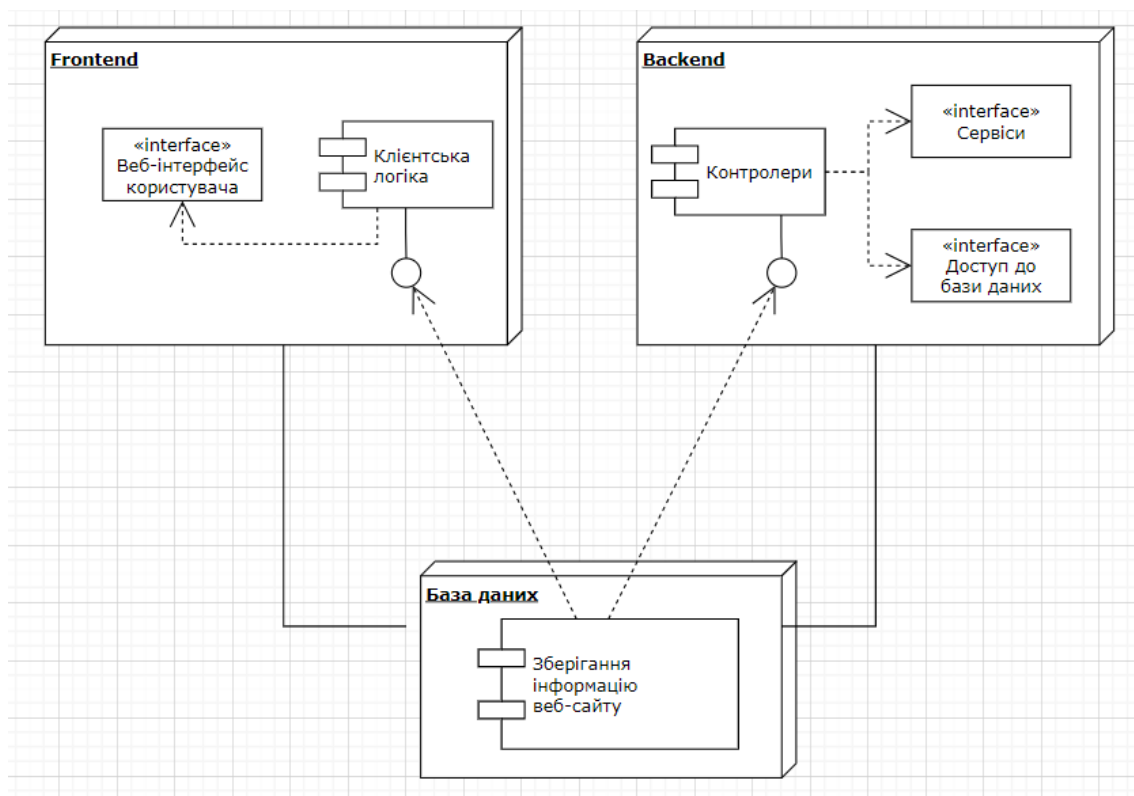


Рис. 3.1 - Діаграма розгортання веб-сайту для пошуку роботи та працівників

## Опис компонентів:

### Frontend:

- Веб-інтерфейс користувача: Це компонент, що відповідає за взаємодію з користувачем. Він відображає веб-сторінки та інтерфейси, з якими користувач безпосередньо взаємодіє.
- Клієнтська логіка: Це логіка, яка виконується на стороні клієнта (в браузері користувача). Вона обробляє введення користувача, взаємодіє з веб-інтерфейсом, і може надсилати запити до бекенду.

### Backend:

- Контролери: Це компоненти, що обробляють запити, які надходять від фронтенду. Вони визначають, яку дію потрібно виконати, і взаємодіють з іншими сервісами або базою даних.
- Сервіси: Це логіка на стороні сервера, яка обробляє бізнес-логіку. Сервіси можуть виконувати складні операції, обробляти дані та взаємодіяти з іншими компонентами системи.
- Доступ до бази даних: Це інтерфейс для взаємодії з базою даних, що дозволяє зберігати, оновлювати та отримувати дані, необхідні для роботи веб-сайту.

### База даних:

- Зберігання інформації веб-сайту: Це компонент, що відповідає за збереження всіх даних веб-сайту, включаючи інформацію про користувачів, вакансії, відгуки на вакансії та інші дані. Взаємодіє з бекендом для збереження та отримання даних.

### Взаємодія компонентів:

- Фронтенд і Бекенд: Фронтенд надсилає запити до бекенду, використовуючи клієнтську логіку, яка взаємодіє з контролерами бекенду. Контролери обробляють ці запити, виконуючи необхідні операції за допомогою сервісів та інтерфейсів доступу до бази даних.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 53   |

- Бекенд і База даних: Контролери бекенду через інтерфейси доступу до бази даних взаємодіють з базою даних для збереження або отримання необхідної інформації.
- Фронтенд і База даних: Пряма взаємодія між фронтендом та базою даних відсутня. Всі операції з базою даних здійснюються через бекенд.

Ця діаграма допомагає зрозуміти, як різні частини системи працюють разом для забезпечення функціональності веб-сайту для пошуку роботи та працівників.

### 3.2 Структура інтерфейсу веб-сайту для пошуку роботи та працівників

У цьому пункті ми розглянемо структуру інтерфейсу веб-сайту для пошуку роботи та працівників. Веб-інтерфейс є ключовим елементом системи, оскільки він забезпечує взаємодію між користувачами (роботодавцями та працівниками) та функціональністю веб-сайту. Зручний та інтуїтивно зрозумілий інтерфейс сприяє ефективному використанню платформи, покращує користувацький досвід (UX) та підвищує загальну задоволеність користувачів.

Розглянемо основні компоненти інтерфейсу, такі як головна сторінка, сторінки публічного профілю працівників, пошук вакансій та працівників, сторінки вакансій, а також інші важливі елементи навігації та взаємодії. Детальний аналіз структури інтерфейсу дозволить зрозуміти, як організовано інформацію та функціональні можливості веб-сайту, а також як користувачі можуть взаємодіяти з цими елементами для досягнення своїх цілей.

#### Головна сторінка:

На головній сторінці користувач може базово ознайомитись з інформацією про веб-сайт, та перейти до реєстрації за відповідними кнопкам.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 54   |

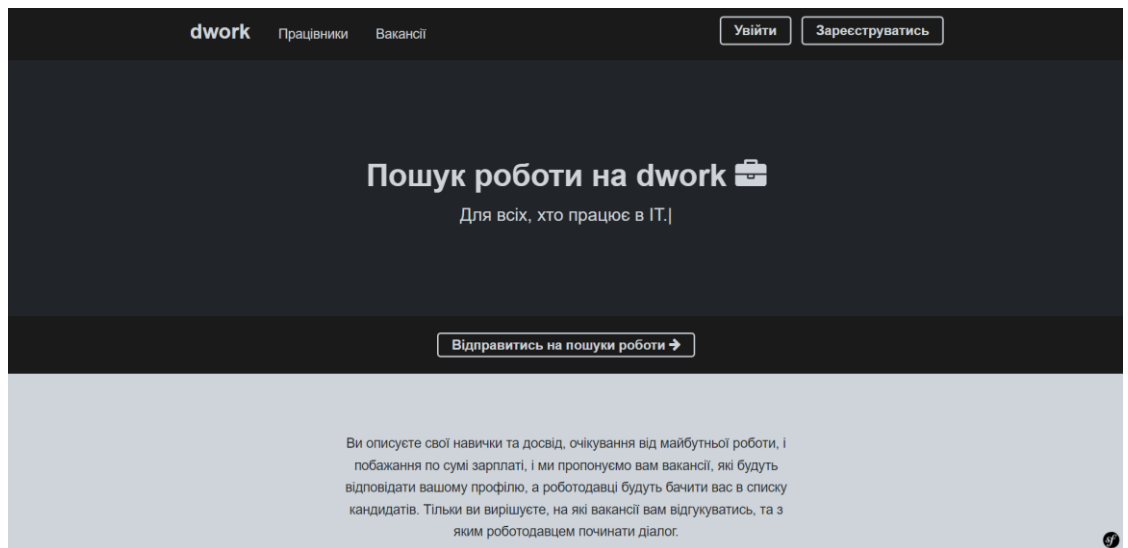


Рис. 3.2 - Головна сторінка



Рис. 3.3 - Головна сторінка

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 55   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

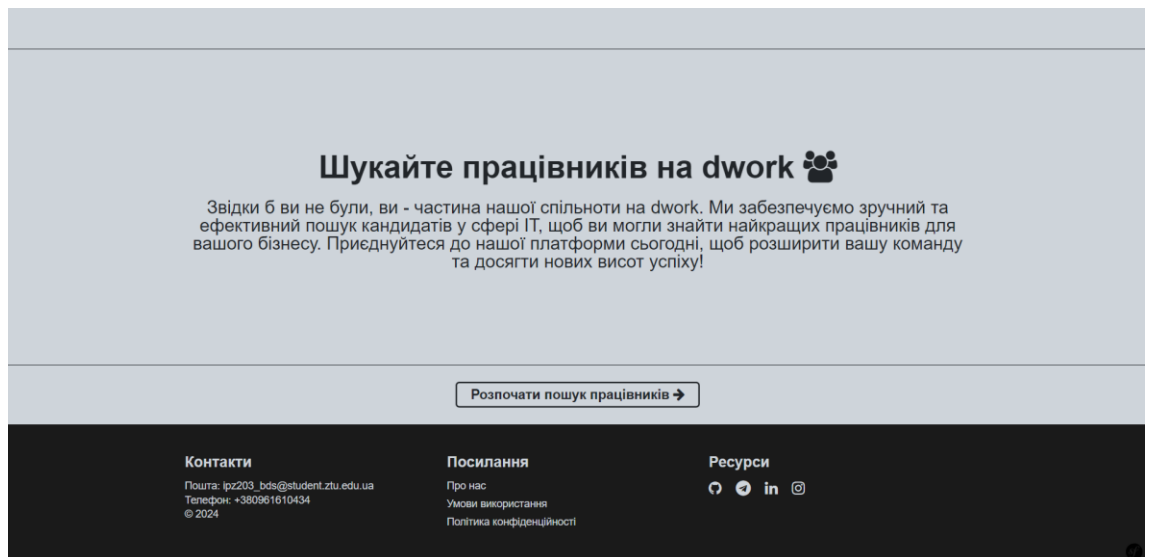


Рис. 3.4 - Головна сторінка

З головної сторінки не авторизований користувач може перейти на сторінку входу, реєстрації, перегляду працівників, або вакансій тому розглянемо їх.

### Сторінка входу:

На сторінці входу користувач може увійти у свій профіль, або перейти на сторінку реєстрації чи використати функцію забули пароль.

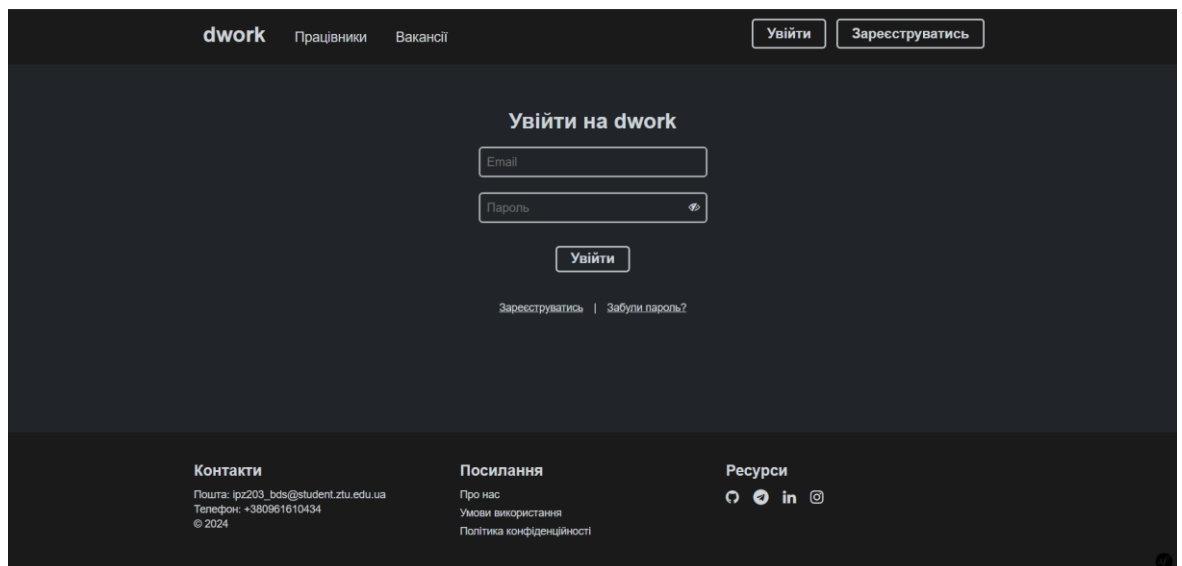


Рис. 3.5 - Сторінка входу

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 56   |

Сторінка реєстрації:

На сторінці реєстрації користувач може обрати тип акаунт, та ввести свої дані.

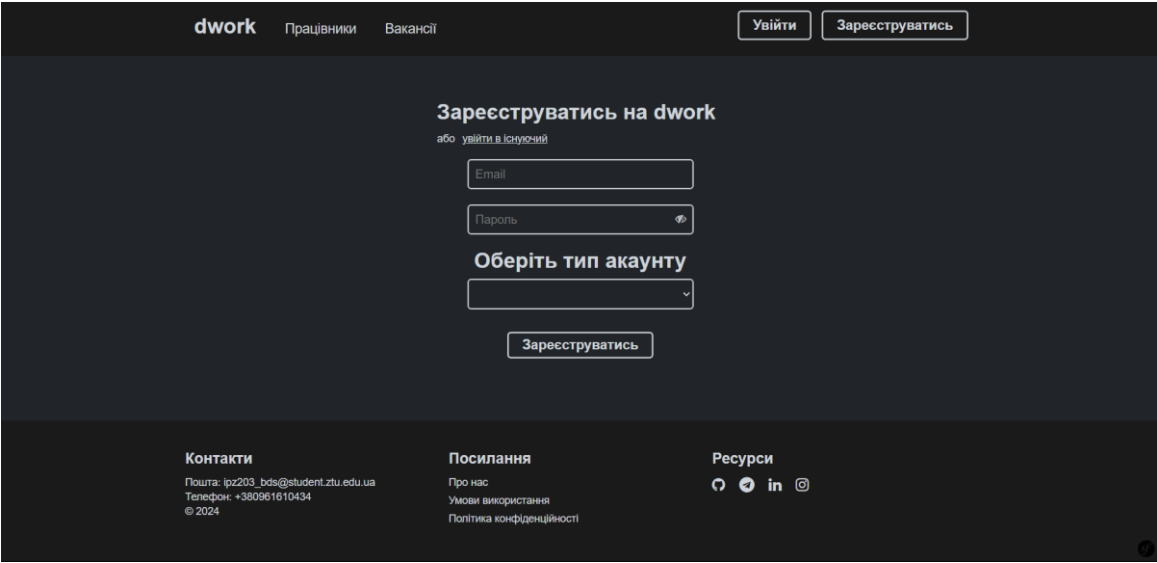


Рис. 3.6 - Сторінка реєстрації

Сторінка працівників:

На сторінці працівників користувач може бачити всіх активних працівників та фільтрувати їх за своїм бажанням.

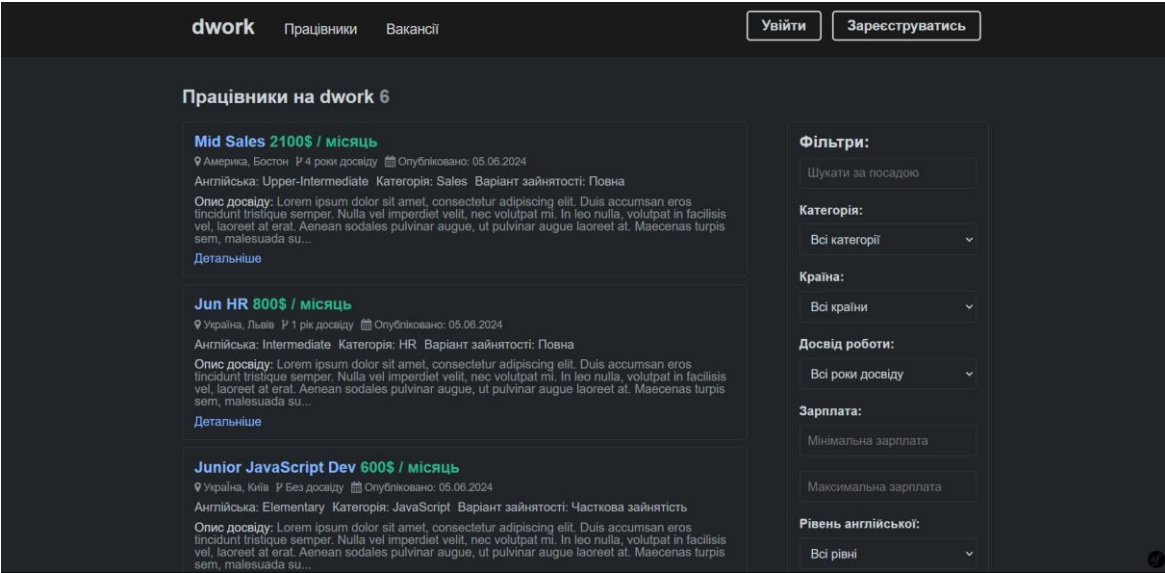


Рис. 3.7 - Сторінка працівників

Сторінка вакансій:

На сторінці вакансій користувач може переглядати список опублікованих вакансій та фільтрувати їх.

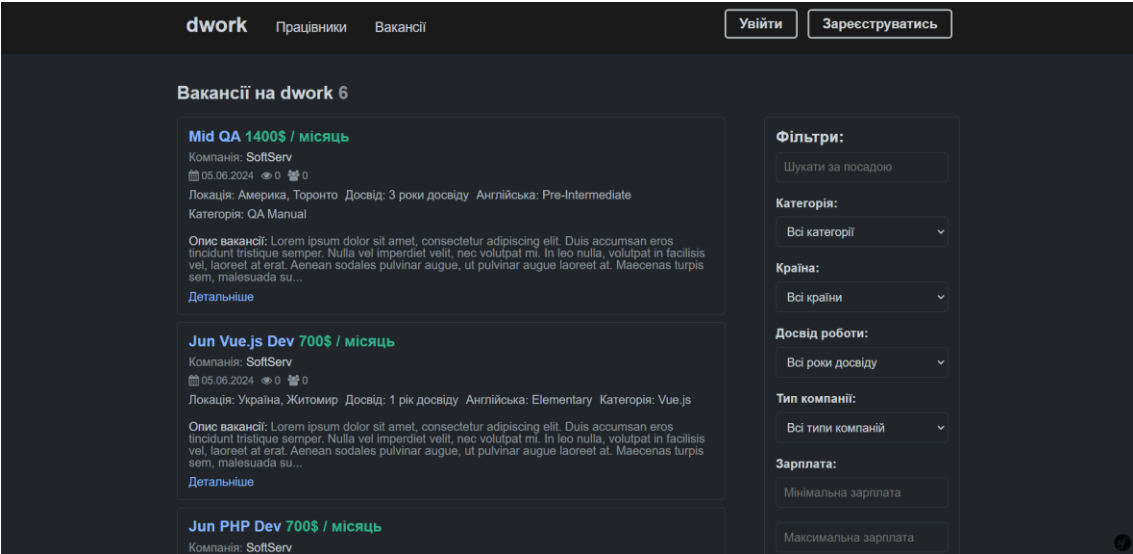


Рис. 3.8 - Сторінка вакансій

Тепер ми розглянемо інтерфейс авторизованого працівника, сторінка перегляду вакансій аналогічна тій яка виводиться і не для авторизованого користувача, тому сенсу акцентувати на ній увагу немає

Сторінка профілю працівника:

На сторінці профілю працівник може заповнити базову інформацію про себе.

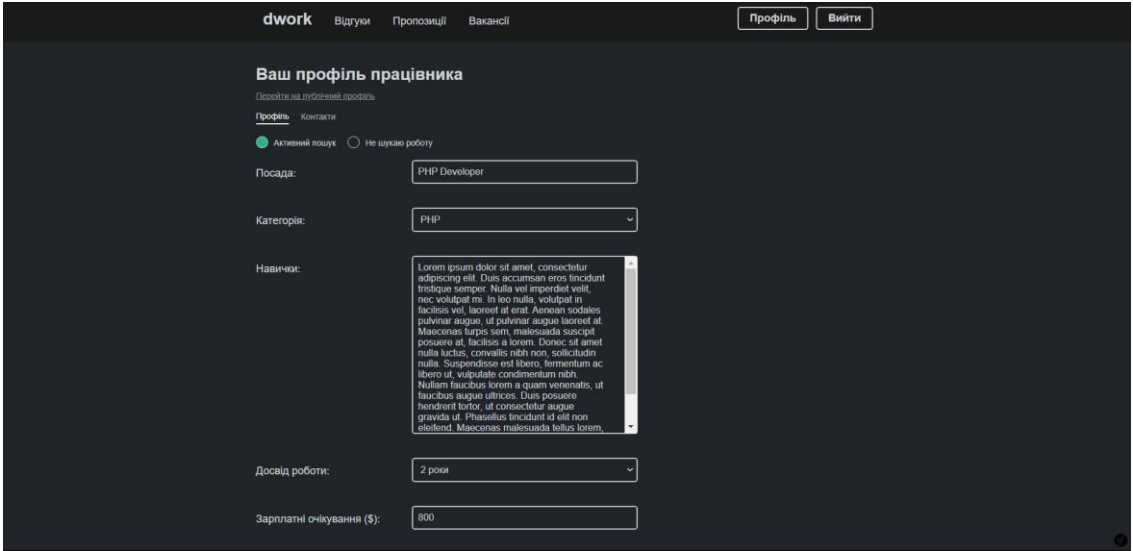


Рис. 3.9 - Сторінка профілю працівника

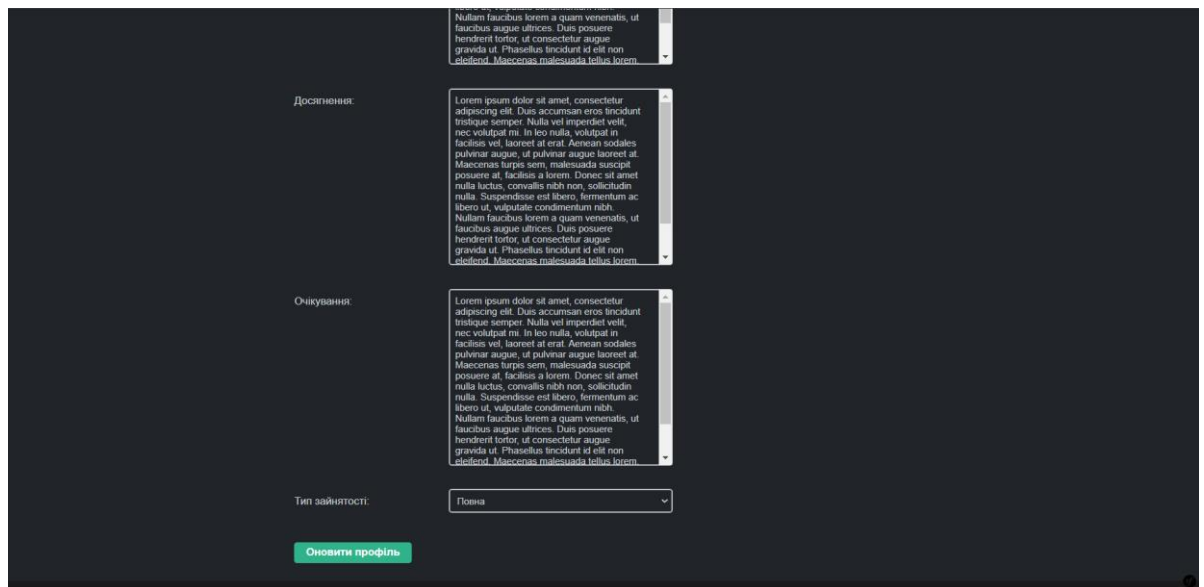


Рис. 3.10 - Сторінка профілю працівника

### Сторінка контактів:

На сторінці контактів працівник може заповнити свої контакти.

Рис. 3.11 - Сторінка контактів працівника

### Сторінка публічного профілю працівника:

На сторінці публічного профілю працівник може побачити як його профіль виглядає для роботодавців.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 59   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

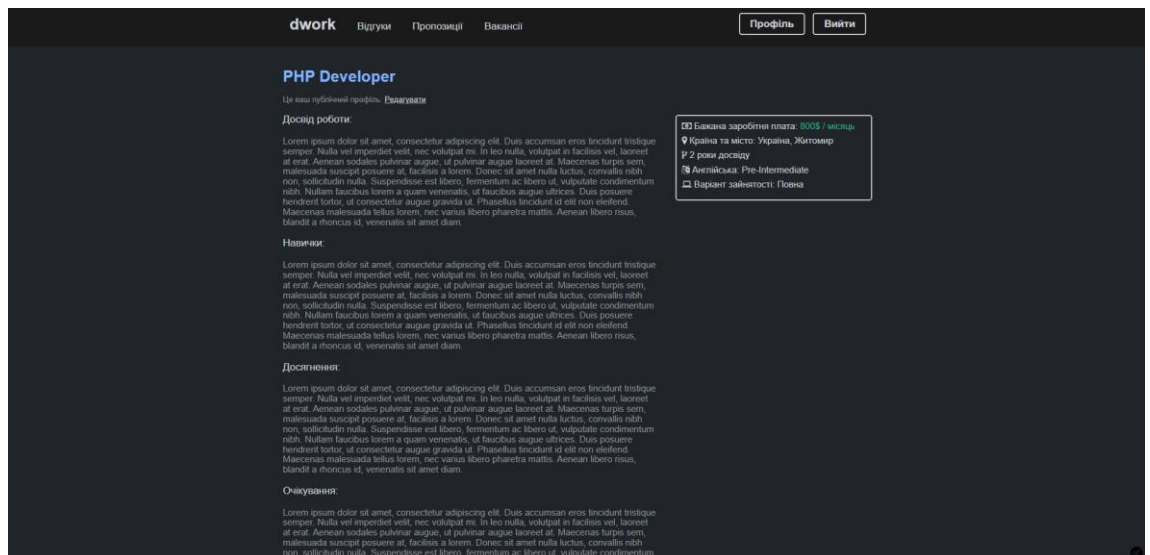


Рис. 3.12 - Сторінка публічного профілю працівника

## Сторінка відгуків на вакансії:

На сторінці відгуків працівник бачить список відгуків на вакансії, та може видаляти їх якщо роботодавець ще не виконав ніяких дій з відгуком.

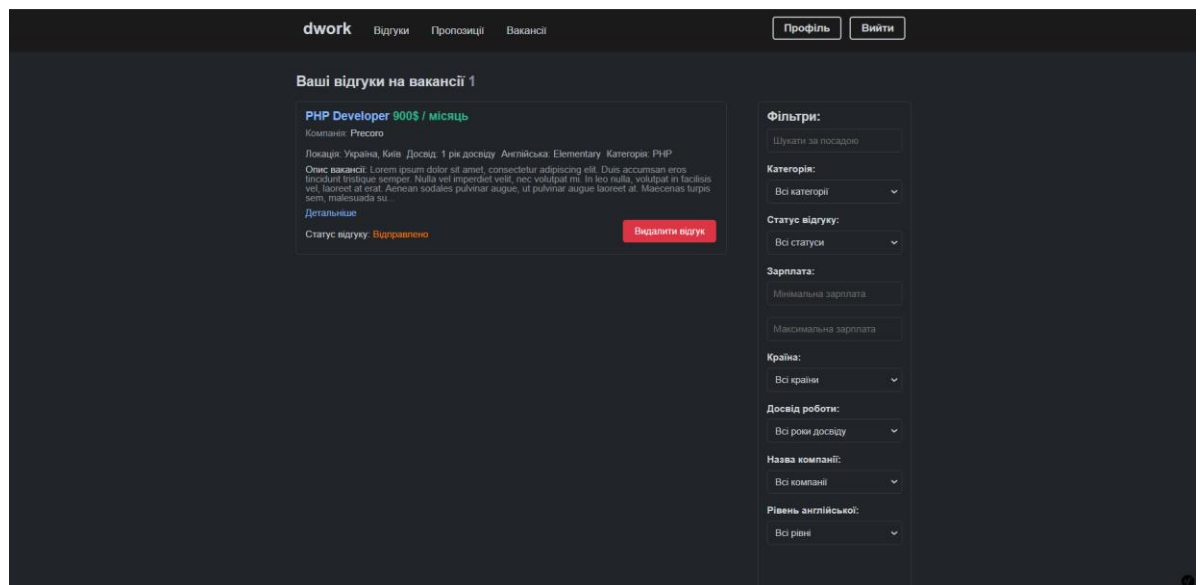


Рис. 3.13 - Сторінка відгуків на вакансії

## Сторінка пропозицій від роботодавців:

На сторінці пропозицій працівник бачить список пропозицій від роботодавців, він може прийняти пропозицію або відхилити, у разі прийняття роботодавець отримає доступ до контактів працівника.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 60   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

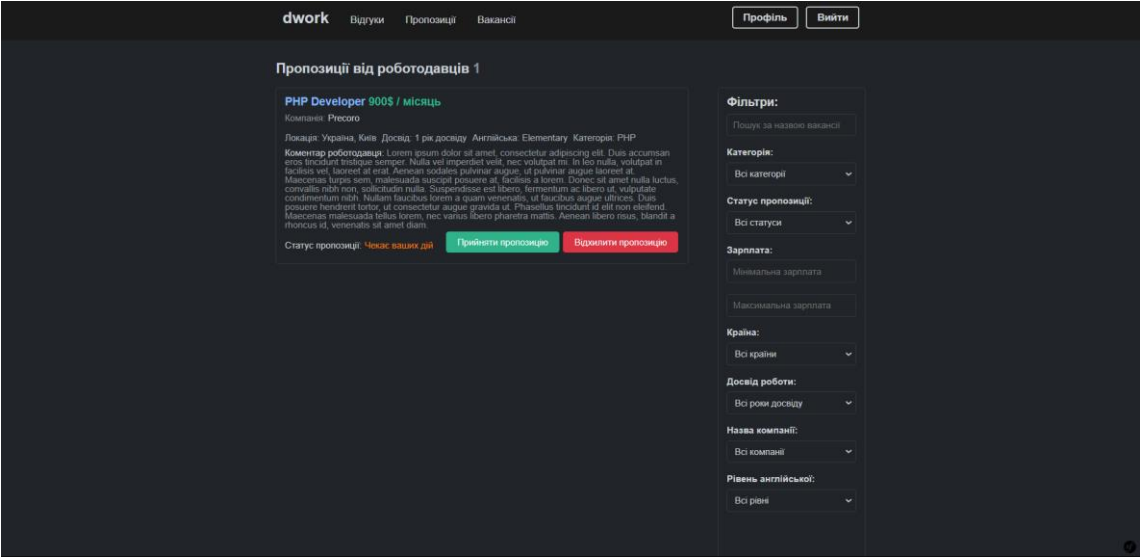


Рис. 3.14 - Сторінка пропозицій від роботодавців

### Сторінка вакансії:

На сторінці вакансії працівник бачить детальну інформацію про вакансію та компанію, та має можливість використати кнопку відгуку на вакансію.

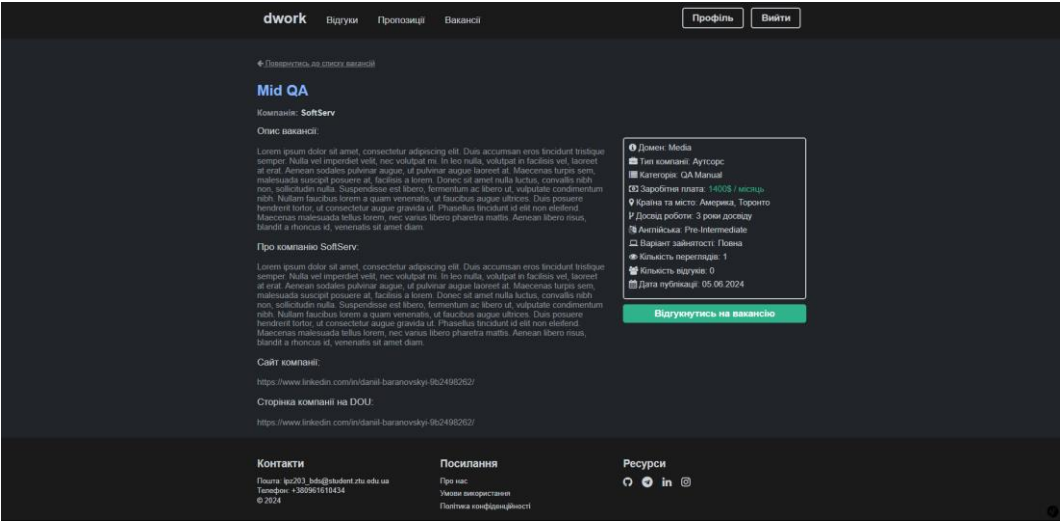


Рис. 3.15 - Сторінка вакансії

У формі відгуку користувач має написати текст для роботодавця.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 61   |



## Сторінка відгуків на вакансії зі сторони роботодавця:

На сторінці відгуків роботодавець має доступ до контактів та резюме працівника який відгукнувся на вакансію, переглянувши їх роботодавець вирішує приймати відгук чи відхилити.

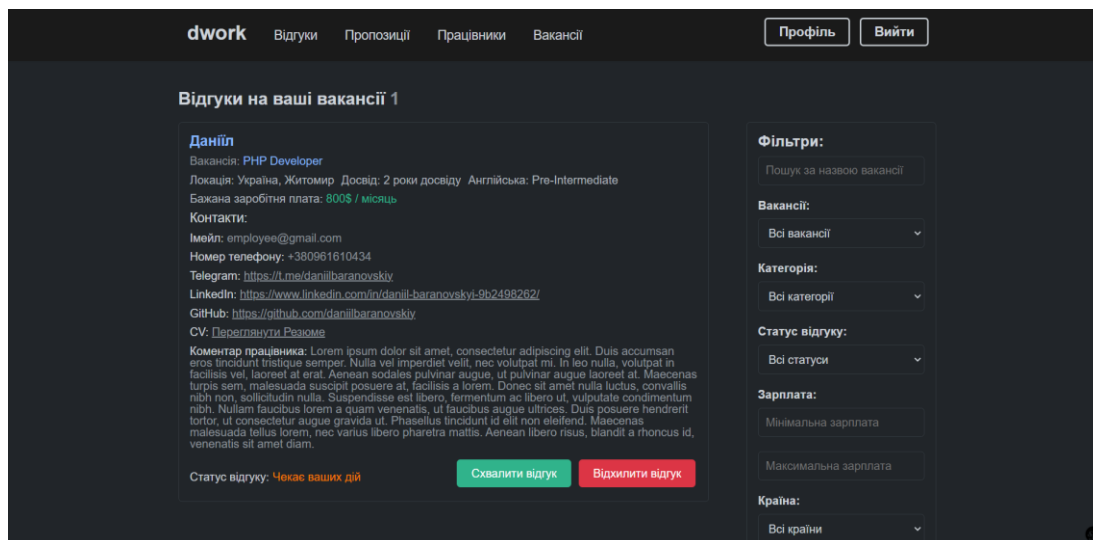


Рис. 3.18 - Сторінка відгуків на вакансії

## Сторінка пропозицій зі сторони роботодавця:

На сторінці пропозицій роботодавець бачить всі пропозиції які він надсилав працівникам і у раз прийняття працівником пропозиції він буде мати доступ до контактів цього працівника.

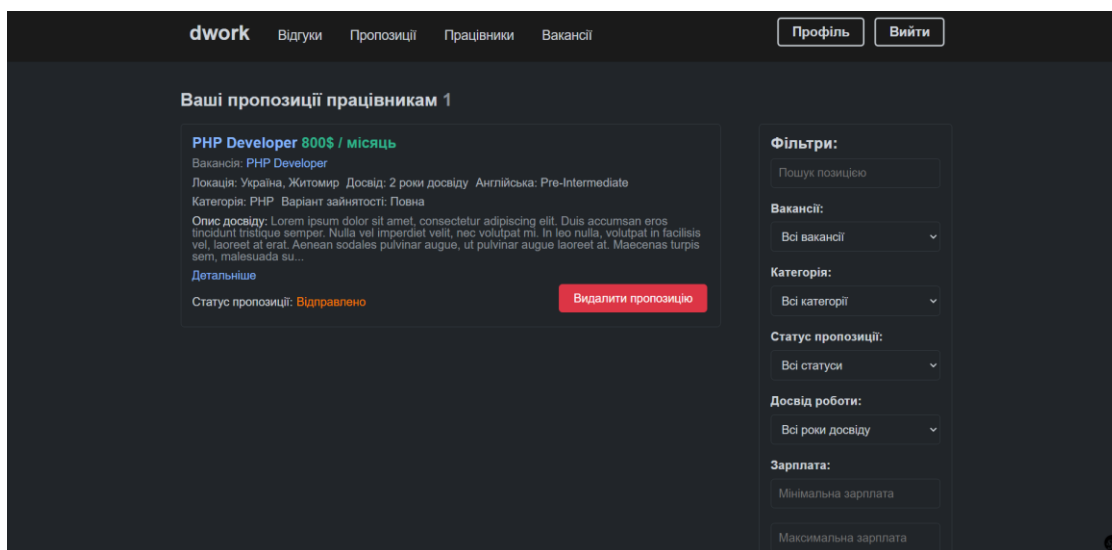


Рис. 3.19 - Сторінка пропозицій

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 63   |

Сторінка вакансій роботодавця:

На сторінці вакансій роботодавець бачить всі свої вакансії та може виконувати відповідні дії з ними.

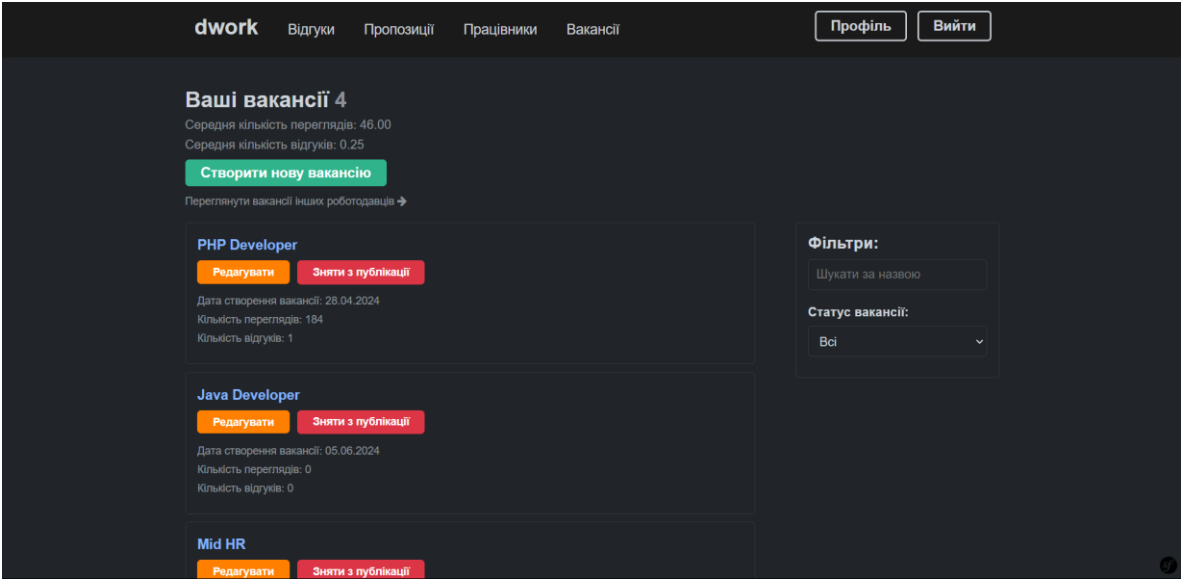


Рис. 3.20 - Сторінка вакансій роботодавця

Сторінка створення вакансії:

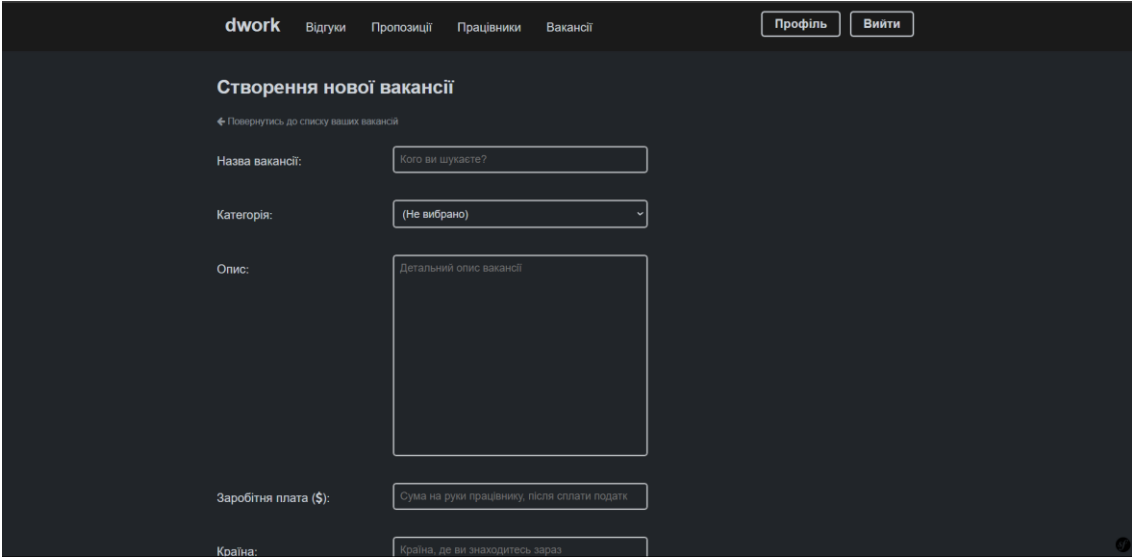


Рис. 3.21 - Сторінка створення вакансії

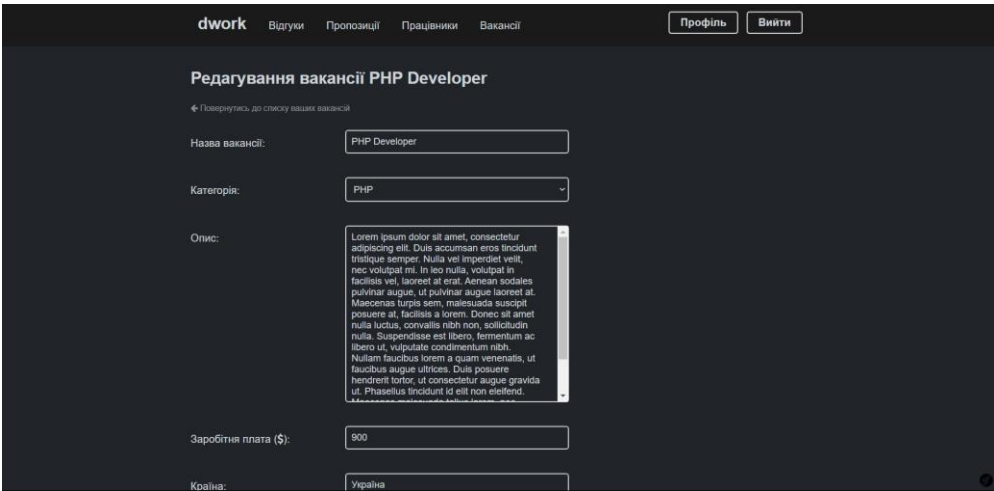


Рис. 3.22 - Сторінка редагування вакансії

Тепер базово розглянемо інтерфейс адмін панелі.

**Головна сторінка адмін панелі:**

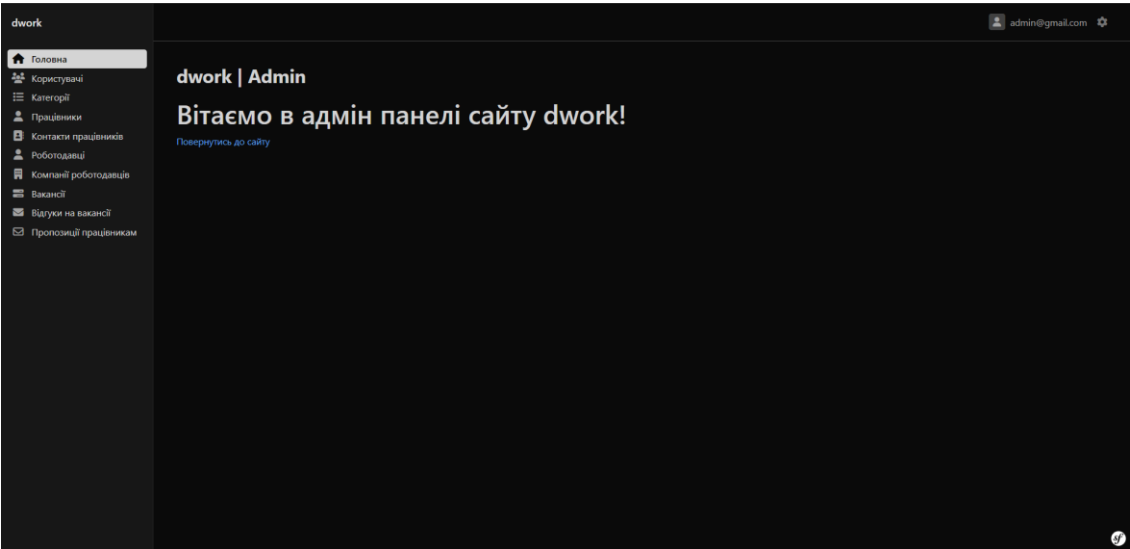


Рис. 3.23 - Головна сторінка адмін панелі

**Сторінка користувачів адмін панелі:**

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 65   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

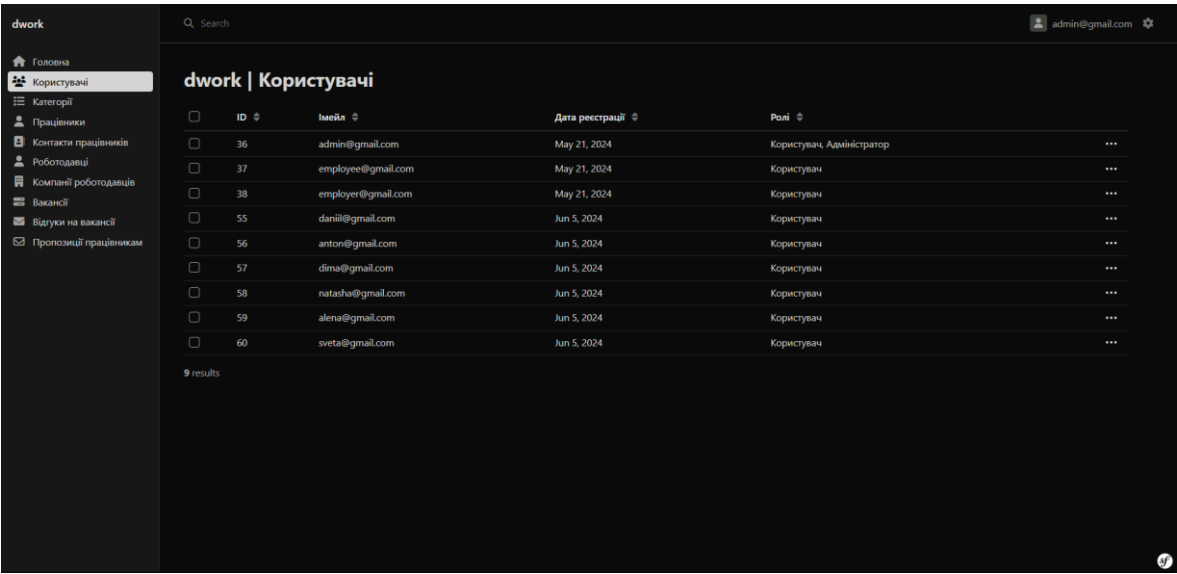


Рис. 3.24 - Сторінка користувачів адмін панелі

Сторінка категорій адмін панелі:

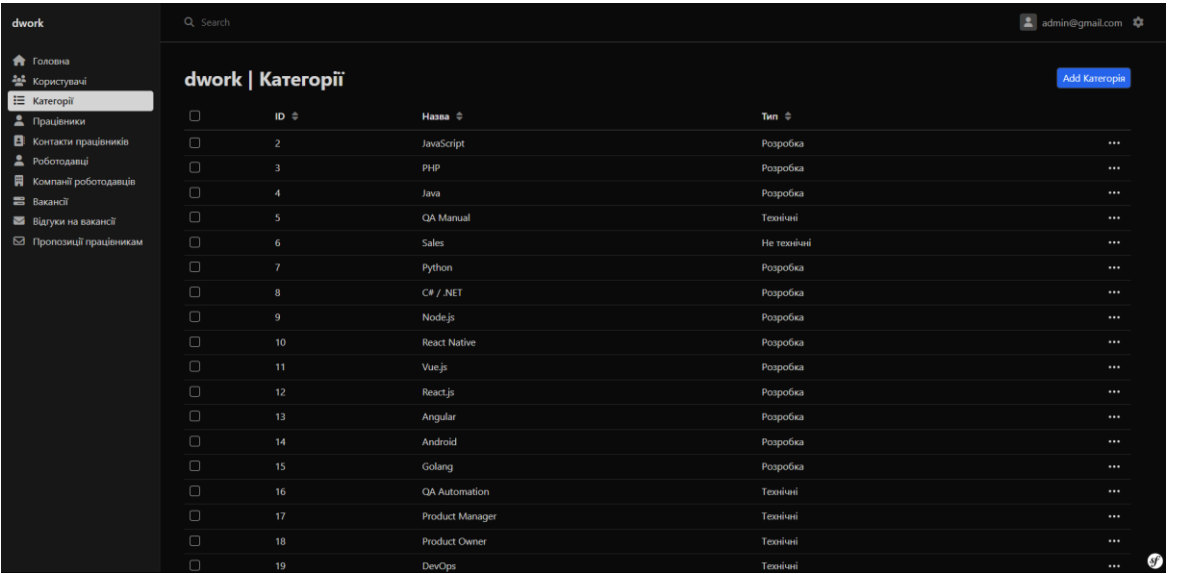


Рис. 3.25 - Сторінка категорій адмін панелі

3.3 Тестування роботи веб-сайту для пошуку роботи та працівників

Тестування веб-сайту для пошуку роботи та працівників є важливим етапом розробки, що забезпечує високу якість та надійність кінцевого продукту. У цьому підпункті ми розглянемо процес мануального тестування, який допомагає виявити та усунути помилки, забезпечити коректну роботу всіх

функціональних можливостей та підтвердити відповідність веб-сайту вимогам користувачів.

Мною було мануально протестовано секюріті частину сайту і перевірено наскільки добре відловлюються помилки.

**Перехід неавторизованим користувачем в адмін панель:**

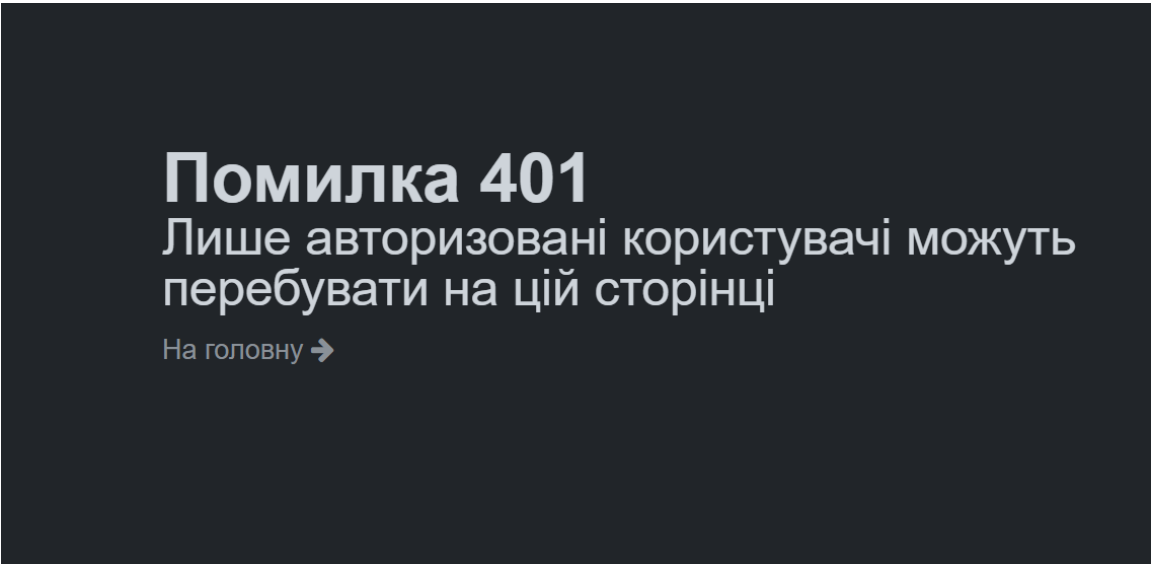


Рис. 3.26 - Сторінка помилки 401

**Перехід в адмін панель користувачем без адмін ролі:**

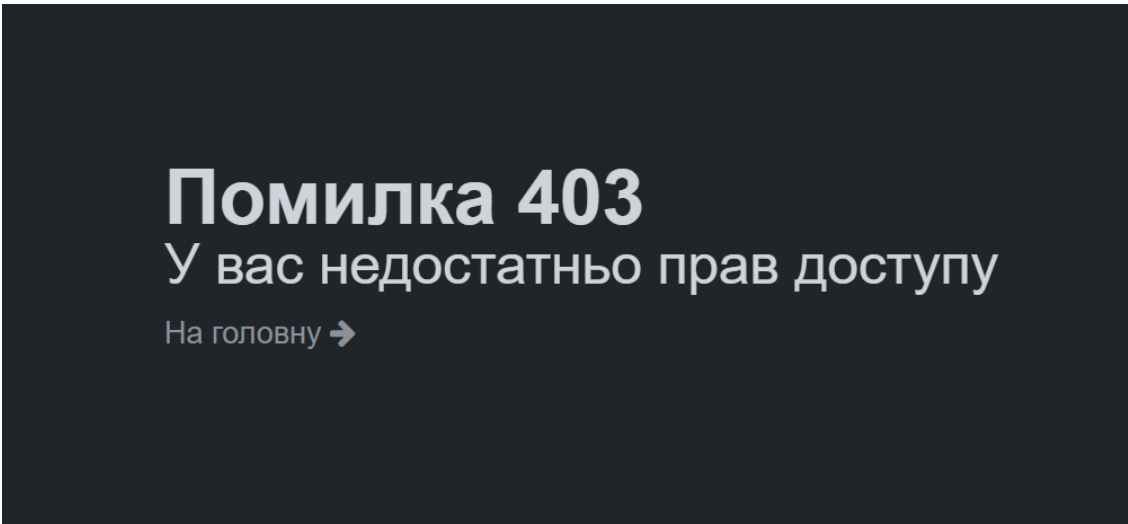


Рис. 3.27 - Сторінка помилки 403

Перехід на неіснуючий роут:

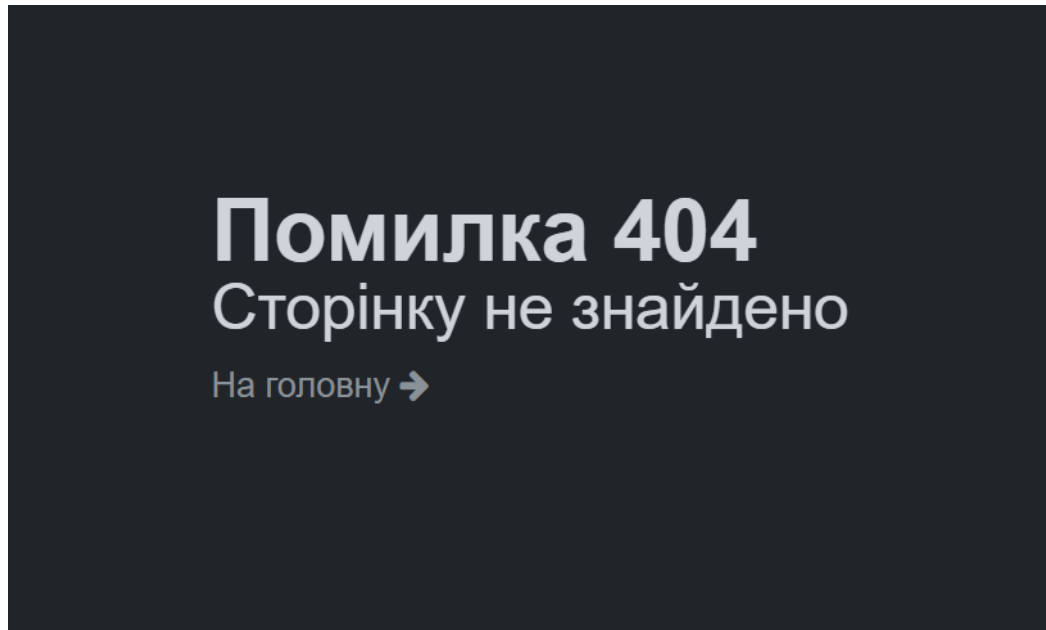


Рис. 3.28 - Сторінка помилки 404

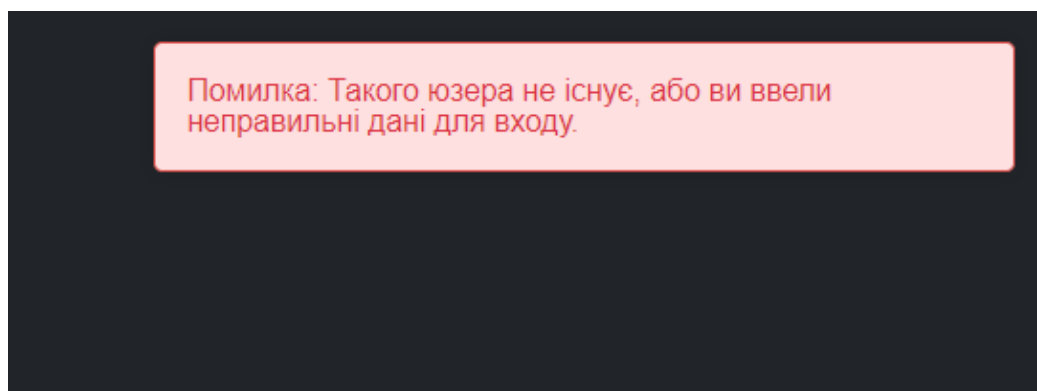


Рис. 3.29 - Коректний вивід помилки під час входу

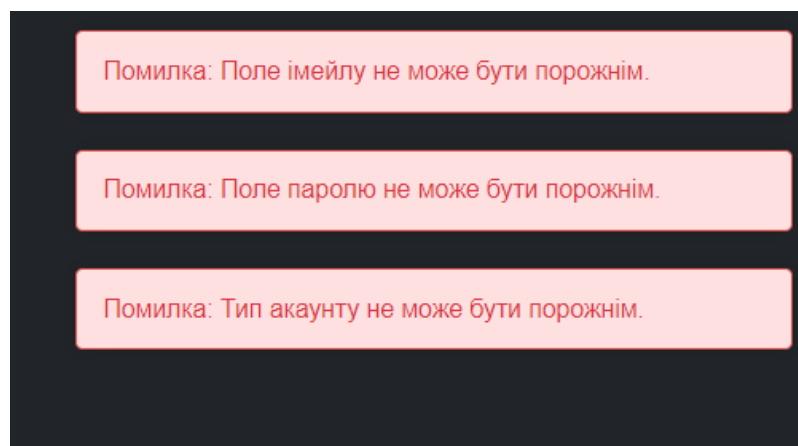


Рис. 3.30 - Коректний вивід помилки під час реєстрації

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 68   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

На цих базових прикладах було представлено що сайт відловлює помилки та відкриває відповідні сторінки з інформацією про помилку та виводить відповідні помилки валідації [20].

### Висновки до третього розділу

У третьому розділі ми детально розглянули ключові аспекти, пов'язані з інтерфейсом та порядком роботи з веб-сайтом для пошуку роботи та працівників. Порядок встановлення та налаштування параметрів системи є комплексним процесом, який включає не лише налаштування окремих компонентів, але й інтеграцію системи в цілому. Діаграма розгортання допомагає зрозуміти структуру і взаємозв'язки між компонентами системи, що сприяє ефективному налаштуванню і стабільній роботі платформи.

Зручний та інтуїтивно зрозумілий інтерфейс є критично важливим для забезпечення позитивного користувацького досвіду. Дизайн інтерфейсу повинен враховувати потреби різних категорій користувачів, зокрема роботодавців та кандидатів, забезпечуючи їм легкий доступ до необхідної функціональності.

Мануальне тестування є важливим етапом у процесі розробки, що допомагає виявити можливі помилки та невідповідності, забезпечуючи надійність і якість веб-сайту. Проведення тестування за різними сценаріями використання дозволяє переконатися в тому, що всі функції веб-сайту працюють коректно і відповідають очікуванням користувачів.

Загалом, третій розділ акцентує увагу на важливості правильного налаштування системи, створення зручного інтерфейсу та ретельного тестування для забезпечення високоякісного та надійного веб-сайту для пошуку роботи та працівників. Виконання цих етапів забезпечує не тільки технічну стабільність, але й позитивний користувацький досвід, що є ключовим для успіху платформи.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    |      |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    | 69   |

## ВИСНОВКИ

У даній кваліфікаційній роботі було успішно розроблено веб-сайт для пошуку роботи та працівників, що сприяє ефективній взаємодії між роботодавцями та потенційними кандидатами.

Перш за все, було визначено вимоги до системи та розроблено об'єктно-орієнтовану структуру, що враховує потреби користувачів. Це дозволило забезпечити високий рівень організації та функціональності веб-сайту.

На етапі розробки бази даних було використано MySQL, що дозволило забезпечити надійне зберігання та ефективне управління інформацією. Це є критично важливим для платформи, яка обробляє значні обсяги даних, пов'язаних з вакансіями та кандидатами.

Ретельне проектування та реалізація алгоритмів роботи системи гарантували надійність та зручність використання веб-сайту. Ключові алгоритми, такі як реєстрація користувачів, пошук вакансій та відгуки на них, були детально опрацьовані та впроваджені.

Важливою частиною роботи було проектування інтерфейсу, що забезпечує інтуїтивно зрозумілий та зручний користувацький досвід. Інтерфейс був розроблений з урахуванням сучасних вимог до UI/UX дизайну, що робить платформу привабливою та зручною для користувачів.

Загалом, виконання цієї кваліфікаційної роботи дозволило створити функціональний та зручний веб-сайт для пошуку роботи та працівників. Розроблена платформа забезпечує ефективну взаємодію між роботодавцями та працівниками, сприяє швидкому знаходженню відповідних вакансій та працівників, що в свою чергу підвищує загальну ефективність процесу найму.

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 70   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформація про Symfony [Електронний ресурс] – Режим доступу до ресурсу:  
<https://symfony.com/doc/current/index.html>
2. Інформація про Vue.js [Електронний ресурс] – Режим доступу до ресурсу:  
<https://vuejs.org/index.html>
3. Інформація про налаштування Vue.js в Symfony [Електронний ресурс] – Режим доступу до ресурсу:  
<https://symfony.com/doc/6.4/frontend/encore/vuejs.html>
4. Інформація про MVC патерн [Електронний ресурс] – Режим доступу до ресурсу: <https://highload.today/uk/blogs/shho-take-mvc-ta-mvp-patterni/>
5. Інформація про RESTful API [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/shcho-take-rest-api/>
6. Інформація про сутності в Symfony [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/doctrine.html>
7. Інформація про контролери в Symfony [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/controller.html>
8. Інформація про патерни проектування [Електронний ресурс] – Режим доступу до ресурсу:  
<https://refactoring.guru/uk/design-patterns>
9. Інформація про PhpStorm [Електронний ресурс] – Режим доступу до ресурсу: <https://itpro.ua/product/jetbrains-phpstorm/?tab=description>
10. Інформація про Webpack [Електронний ресурс] – Режим доступу до ресурсу: <https://webpack.js.org/>
11. Інформація про SCSS [Електронний ресурс] – Режим доступу до ресурсу:  
<https://sass-lang.com/documentation/style-rules/>
12. Інформація про MySQL [Електронний ресурс] – Режим доступу до ресурсу:  
<https://dev.mysql.com/doc/>
13. Інформація про Beekeeper Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://www.beekeeperstudio.io/about>

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 71   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

14. Інформація про Symfony Server [Електронний ресурс] – Режим доступу до ресурсу: [https://symfony.com/doc/current/setup/symfony\\_server.html](https://symfony.com/doc/current/setup/symfony_server.html)
15. Інформація про PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>
16. Інформація про JS [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.javascript.info/intro>
17. Інформація про форми в Symfony [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/forms.html>
18. Інформація про надсилання імейлів в Symfony [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/mailer.html>
19. Інформація про twig [Електронний ресурс] – Режим доступу до ресурсу: <https://twig.symfony.com/>
20. Інформація про валідацію в Symfony [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/validation.html>

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 72   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

# ДОДАТКИ

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 73   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## Лістинг коду VacancyRestController:

```

class VacancyRestController extends AbstractController
{
    /**
     * @var EntityManagerInterface
     */
    private EntityManagerInterface $entityManager;

    /**
     * @param EntityManagerInterface $entityManager
     */
    public function __construct(EntityManagerInterface $entityManager)
    {
        $this->entityManager = $entityManager;
    }

    /**
     * @throws \Exception
     */
    #[Route('/app/vacancy', name: 'app_add_vacancy', methods: ['POST'])]
    public function addVacancyAction(Request $request): Response
    {
        $vacancy = new Vacancy();
        $user = $this->getUser();
        $employer = $user->getEmployer();
        $requestData = json_decode($request->getContent(), true);
        $form = $this->createForm(VacancyType::class, $vacancy);
        $form->submit($requestData);

        if ($form->isSubmitted() && $form->isValid()) {
            if ($requestData['jobCategory'] === null) {
                $vacancy->setJobCategory(null);
            } else {
                $jobCategory = $this->entityManager->getRepository(JobCategory::class)->findOneBy(['id' =>
                $requestData['jobCategory']['id']]);
                $vacancy->setJobCategory($jobCategory);
            }

            $vacancy->setEmployer($employer);
            $this->entityManager->persist($vacancy);
            $this->entityManager->flush();

            return new JsonResponse($vacancy, Response::HTTP_CREATED);
        }

        $formErrors = $form->getErrors(true);
        $errors = [];

        foreach ($formErrors as $error) {
            $errors[] = $error->getMessage();
        }

        return new JsonResponse(['errors' => $errors],
        Response::HTTP_BAD_REQUEST);
    }

    /**
     * @throws \Exception

```

|      |      |                   |        |      |                    |      |
|------|------|-------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С. |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.       |        |      |                    | 74   |
| Змн. | Арк. | № докум.          | Підпис | Дата |                    |      |

```

    */
    #[Route('/app/vacancy/{id}', name: 'app_edit_vacancy', methods: ['PATCH'])]
    public function editVacancyAction(Request $request, int $id): Response
    {
        $vacancy = $this->entityManager->getRepository(Vacancy::class)-
>findOneBy(['id' => $id]);
        if (!$vacancy) {
            throw $this->createNotFoundException();
        }

        $requestData = json_decode($request->getContent(), true);

        $form = $this->createForm(VacancyType::class, $vacancy);
        $form->submit($requestData);

        if ($form->isSubmitted() && $form->isValid()) {
            if ($requestData['jobCategory'] === null) {
                $vacancy->setJobCategory(null);
            } else {
                $jobCategory = $this->entityManager-
>getRepository(JobCategory::class)->findOneBy(['id' =>
$requestData['jobCategory']['id']]);
                $vacancy->setJobCategory($jobCategory);
            }

            $this->entityManager->persist($vacancy);
            $this->entityManager->flush();

            return new JsonResponse($vacancy, Response::HTTP_OK);
        }

        $formErrors = $form->getErrors(true);
        $errors = [];

        foreach ($formErrors as $error) {
            $errors[] = $error->getMessage();
        }

        return new JsonResponse(['errors' => $errors],
Response::HTTP_BAD_REQUEST);
    }

    #[Route('/app/vacancy/unpublish/{id}', name: 'app_vacancy_unpublish',
methods: ['PATCH'])]
    public function unpublishVacancyAction(int $id): Response
    {
        $vacancy = $this->entityManager->getRepository(Vacancy::class)-
>findOneBy(['id' => $id]);
        if (!$vacancy) {
            throw $this->createNotFoundException();
        }

        $vacancy->setEnable(false);

        $this->entityManager->persist($vacancy);
        $this->entityManager->flush();

        return new JsonResponse($vacancy, Response::HTTP_OK);
    }

```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 75   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

```

        #[Route('/app/vacancy/publish/{id}', name: 'app_vacancy_publish', methods:
['PATCH'])]
        public function publishVacancyAction(int $id): Response
        {
            $vacancy = $this->entityManager->getRepository(Vacancy::class)-
>findOneBy(['id' => $id]);
            if (!$vacancy) {
                throw $this->createNotFoundException();
            }

            $vacancy->setEnable(true);

            $this->entityManager->persist($vacancy);
            $this->entityManager->flush();

            return new JsonResponse($vacancy, Response::HTTP_OK);
        }

        #[Route('/app/vacancy/delete/{id}', name: 'app_vacancy_delete', methods:
['DELETE'])]
        public function deleteVacancyAction(int $id): Response
        {
            $vacancy = $this->entityManager->getRepository(Vacancy::class)-
>findOneBy(['id' => $id]);
            if (!$vacancy) {
                throw $this->createNotFoundException();
            }

            $this->entityManager->remove($vacancy);
            $this->entityManager->flush();

            return new Response(null, Response::HTTP_NO_CONTENT);
        }

```

Додаток В

### Лістинг коду EmployeeRestController:

```

class EmployeeRestController extends AbstractController
{
    /**
     * @var EntityManagerInterface
     */
    private EntityManagerInterface $entityManager;

    /**
     * @var string
     */
    private string $resumeDirectory;

    /**
     * @param EntityManagerInterface $entityManager
     * @param string $resumeDirectory
     */
    public function __construct(EntityManagerInterface $entityManager, string
$resumeDirectory)
    {
        $this->entityManager = $entityManager;
        $this->resumeDirectory = $resumeDirectory;
    }

```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 76   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

```

/**
 * @throws \Exception
 */
#[Route('/app/employee/{id}', name: 'app_edit_employee', methods:
['PATCH'])]
public function editEmployeeAction(Request $request, int $id): Response
{
    $employee = $this->entityManager->getRepository(Employee::class)-
>findOneBy(['id' => $id]);
    $requestData = json_decode($request->getContent(), true);
    $form = $this->createForm(EmployeeType::class, $employee);
    $form->submit($requestData);
    if ($form->isSubmitted() && $form->isValid()) {
        if ($requestData['jobCategory'] === null) {
            $employee->setJobCategory(null);
        } else {
            $jobCategory = $this->entityManager-
>getRepository(JobCategory::class)->findOneBy(['id' =>
$requestData['jobCategory']['id']]);
            $employee->setJobCategory($jobCategory);
        }

        $this->entityManager->persist($employee);
        $this->entityManager->flush();

        return new JsonResponse($employee, Response::HTTP_CREATED);
    }

    $formErrors = $form->getErrors(true);
    $errors = [];

    foreach ($formErrors as $error) {
        $errors[] = $error->getMessage();
    }

    return new JsonResponse(['errors' => $errors],
Response::HTTP_BAD_REQUEST);
}

/**
 * @throws \Exception
 */
#[Route('/app/employee/contacts/{id}', name: 'app_edit_employee_contacts',
methods: ['PATCH'])]
public function editEmployeeContactsAction(Request $request, int $id,
Filesystem $filesystem): Response
{
    $employeeContacts = $this->entityManager-
>getRepository(EmployeeContacts::class)->findOneBy(['employee' => $id]);
    $requestData = json_decode($request->getContent(), true);

    $form = $this->createForm(EmployeeContactsType::class,
$employeeContacts);
    $form->submit($requestData);

    if ($form->isSubmitted() && $form->isValid()) {
        if (isset($requestData['resumeFileName'])) {
            $oldEmployeeContacts = $this->entityManager->getUnitOfWork()-
>getOriginalEntityData($employeeContacts);
            $oldFileName = $oldEmployeeContacts['resumeFileName'];

```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    |      |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    | 77   |

```

if ($requestData['resumeFileName'] !== $oldFileName && $oldFileName !== null) {

    $filesystem->remove($this->resumeDirectory . '/' .
$oldFileName);
    }
    $employeeContacts->
>setResumeFileName($requestData['resumeFileName']);
    }

    $this->entityManager->persist($employeeContacts);
    $this->entityManager->flush();

    return new JsonResponse($employeeContacts, Response::HTTP_OK);
}

$formErrors = $form->getErrors(true);
$errors = [];

foreach ($formErrors as $error) {
    $errors[] = $error->getMessage();
}

return new JsonResponse(['errors' => $errors],
Response::HTTP_BAD_REQUEST);
}

#[Route('/app/employee/upload-resume', name: 'app_employee_upload_resume',
methods: ['POST'])]
public function uploadResumeAction(Request $request, SluggerInterface
$slugger): Response
{
    $resumeFile = $request->files->get('resumeFileName');

    if (!$resumeFile instanceof UploadedFile) {
        return new JsonResponse(['errors' => ['Помилка: файл не додано.']],
Response::HTTP_BAD_REQUEST);
    }

    if ($resumeFile->getMimeType() !== 'application/pdf') {
        return new JsonResponse(['errors' => ['Помилка: файл має бути у
форматі PDF.']], Response::HTTP_BAD_REQUEST);
    }

    $originalFilename = pathinfo($resumeFile->getClientOriginalName(),
PATHINFO_FILENAME);
    $safeFilename = $slugger->slug($originalFilename);
    $newFilename = $safeFilename . '-' . uniqid() . '.' . $resumeFile-
>guessExtension();

    try {
        $resumeFile->move($this->resumeDirectory, $newFilename);
    } catch (FileException $e) {
        return new JsonResponse(['errors' => ['Помилка: не вдалось
завантажити файл.']], Response::HTTP_INTERNAL_SERVER_ERROR);
    }

    return new JsonResponse(['filename' => $newFilename],
Response::HTTP_CREATED);
}

```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 78   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

```

#[Route('/app/employee/delete-resume/{id}', name: 'app_employee_delete_resume',
methods: ['DELETE'])]
public function deleteResumeAction(int $id, Filesystem $filesystem):
Response
{
    $employeeContacts = $this->entityManager-
>getRepository(EmployeeContacts::class)->findOneBy(['employee' => $id]);

    if (!$employeeContacts) {
        throw $this->createNotFoundException();
    }

    $oldFileName = $employeeContacts->getResumeFileName();
    if (!$oldFileName) {
        return new JsonResponse(['errors' => ['Помилка: резюме не
знайдено.']], Response::HTTP_NOT_FOUND);
    }

    try {
        $filesystem->remove($this->resumeDirectory . '/' . $oldFileName);
    } catch (FileNotFoundException $e) {
        return new JsonResponse(['errors' => ['Помилка: не вдалось знайти
резюме.']], Response::HTTP_INTERNAL_SERVER_ERROR);
    }

    $employeeContacts->setResumeFileName(null);
    $this->entityManager->flush();

    return new JsonResponse([], Response::HTTP_OK);
}

#[Route('/check-resume/{id}', name: 'check_resume')]
public function checkResumeAction(int $id): BinaryFileResponse
{
    $employeeContacts = $this->entityManager-
>getRepository(EmployeeContacts::class)->findOneBy(['employee' => $id]);
    $resumeFileName = $employeeContacts->getResumeFileName();
    $resumeFilePath = $this->resumeDirectory . '/' . $resumeFileName;
    return $this->file($resumeFilePath, $resumeFileName,
ResponseHeaderBag::DISPOSITION_INLINE);
}
}

```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Барановський Д.С |        |      | ІПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Кубрак Ю.О.      |        |      |                    | 79   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |